



Modes of convergence for term graph rewriting

Bahr, Patrick

Published in:
Logical Methods in Computer Science

DOI:
[10.2168/LMCS-8\(2:6\)2012](https://doi.org/10.2168/LMCS-8(2:6)2012)

Publication date:
2012

Document version
Publisher's PDF, also known as Version of record

Document license:
[CC BY-ND](#)

Citation for published version (APA):
Bahr, P. (2012). Modes of convergence for term graph rewriting. *Logical Methods in Computer Science*, 8(2), [6].
[https://doi.org/10.2168/LMCS-8\(2:6\)2012](https://doi.org/10.2168/LMCS-8(2:6)2012)

MODES OF CONVERGENCE FOR TERM GRAPH REWRITING *

PATRICK BAHR

Department of Computer Science, University of Copenhagen (DIKU)
Universitetsparken 1, 2100 Copenhagen, Denmark
URL: <http://www.diku.dk/~paba>
e-mail address: paba@diku.dk

ABSTRACT. Term graph rewriting provides a simple mechanism to finitely represent restricted forms of infinitary term rewriting. The correspondence between infinitary term rewriting and term graph rewriting has been studied to some extent. However, this endeavour is impaired by the lack of an appropriate counterpart of infinitary rewriting on the side of term graphs. We aim to fill this gap by devising two modes of convergence based on a partial order respectively a metric on term graphs. The thus obtained structures generalise corresponding modes of convergence that are usually studied in infinitary term rewriting.

We argue that this yields a common framework in which both term rewriting and term graph rewriting can be studied. In order to substantiate our claim, we compare convergence on term graphs and on terms. In particular, we show that the modes of convergence on term graphs are conservative extensions of the corresponding modes of convergence on terms and are preserved under unravelling term graphs to terms. Moreover, we show that many of the properties known from infinitary term rewriting are preserved. This includes the intrinsic completeness of both modes of convergence and the fact that convergence via the partial order is a conservative extension of the metric convergence.

1. INTRODUCTION

Non-terminating computations are not necessarily undesirable. For instance, the termination of a reactive system would be usually considered a critical failure. Even computations that, given an input x , should produce an output y are not necessarily terminating in nature either. For example, the various iterative approximation algorithms for π produce approximations of increasing accuracy without ever terminating with the exact value of π . While such iterative approximation computations might not reach the exact target value, they are able to come arbitrary close to the correct value within finite time.

It is this kind of non-terminating computations which is the subject of infinitary term rewriting [24]. It extends the theory of term rewriting by giving a meaning to transfinite reductions instead of dismissing them as undesired and meaningless artifacts. Following the paradigm of iterative approximations, the result of a transfinite reduction is simply the

1998 ACM Subject Classification: F.4.2, F.1.1.

Key words and phrases: term graphs, partial order, metric, infinitary rewriting, graph rewriting.

* Parts of this paper have appeared in the proceedings of RTA 2011 [8].

term that is approximated by the reduction. In general, such a result term can be infinite. For example, starting from the term $rep(0)$, the rewrite rule $rep(x) \rightarrow x :: rep(x)$ produces a reduction

$$rep(0) \rightarrow 0 :: rep(0) \rightarrow 0 :: 0 :: rep(0) \rightarrow 0 :: 0 :: 0 :: rep(0) \rightarrow \dots$$

that approximates the infinite term $0 :: 0 :: 0 :: \dots$. Here, we use $::$ as a binary symbol that we write infix and assume to associate to the right. That is, the term $0 :: 0 :: rep(0)$ is parenthesised as $0 :: (0 :: rep(0))$. Think of the $::$ symbol as the list constructor *cons*.

Term graphs, on the other hand, allow us to explicitly represent and reason about sharing and recursion [3] by dropping the restriction to a tree structure, which we have for terms. Apart from that, term graphs also provide a finite representation of certain infinite terms, viz. *rational terms*. As Kennaway et al. [23, 26] have shown, this can be leveraged in order to finitely represent restricted forms of infinitary term rewriting using *term graph rewriting*.

In this paper, we extend the theory of infinitary term rewriting to the setting of term graphs. To this end, we devise modes of convergence that constrain reductions of transfinite length in a meaningful way. Our approach to convergence is twofold: we generalise the metric on terms that is used to define convergence for infinitary term rewriting [14] to term graphs. In a similar way, we generalise the partial order on terms that has been recently used to define a closely related notion of convergence for infinitary term rewriting [7]. The use of two different – but on terms closely related – approaches to convergence will allow us both to assess the appropriateness of the resulting infinitary calculi and to compare them against the corresponding infinitary calculi of term rewriting.

1.1. Motivation.

1.1.1. Lazy Evaluation. Term rewriting is a useful formalism for studying declarative programs, in particular, functional programs. A functional program essentially consists of functions defined by a set of equations and an expression that is supposed to be evaluated according to these equations. The conceptual process of evaluating an expression is nothing else than term rewriting.

A particularly interesting feature of modern functional programming languages, such as Haskell [29], is the ability to use *conceptually* infinite computations and data structures. For example, the following definition of a function **from** constructs for each number n the infinite list of consecutive numbers starting from n :

$$\mathbf{from}(n) = n :: \mathbf{from}(s(n))$$

Here, we use the binary infix symbol $::$ to denote the list constructor *cons* and **s** for the successor function. While we cannot use the infinite list generated by **from** directly – the evaluation of an expression of the form **from** n does not terminate – we can use it in a setting in which we only read a finite prefix of the infinite list conceptually defined by **from**. Functional languages such as Haskell allow this use of semantically infinite data structures through a *non-strict evaluation* strategy, which delays the evaluation of a subexpression until its result is actually required for further evaluation of the expression. This non-strict semantics is not only a conceptual neatness but in fact one of the major features that make functional programs highly modular [18].

The above definition of the function **from** can be represented as a term rewriting system with the following rule:

$$\text{from}(x) \rightarrow x :: \text{from}(s(x))$$

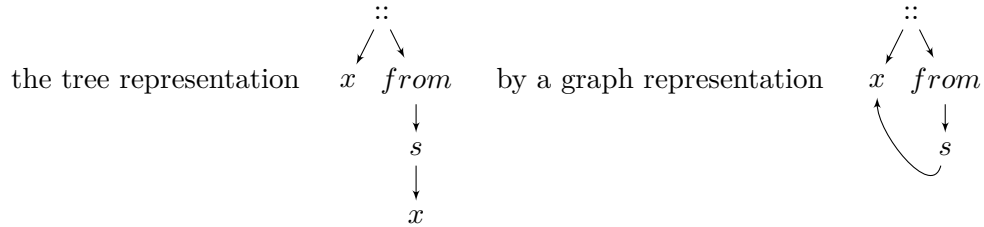
Starting with the term $\text{from}(0)$, we then obtain the following infinite reduction:

$$\text{from}(0) \rightarrow 0 :: \text{from}(s(0)) \rightarrow 0 :: s(0) :: \text{from}(s(s(0))) \rightarrow \dots$$

Infinitary term rewriting [24] provides a notion of convergence that may assign a meaningful result term to such an infinite reduction provided there exists one. In this sense, the above reduction converges to the infinite term $0 :: s(0) :: s(s(0)) :: \dots$, which represents the infinite list of numbers $0, 1, 2, \dots$. Due to this extension of term rewriting with explicit limit constructions for non-terminating reductions, infinitary term rewriting allows us to directly reason about non-terminating functions and infinite data structures.

Non-strict evaluation is rarely found unescorted, though. Usually, it is implemented as *lazy evaluation* [17], which complements a non-strict evaluation strategy with *sharing*. The latter avoids duplication of subexpressions by using pointers instead of copying. For example, the function **from** above duplicates its argument n – it occurs twice on the right-hand side of the defining equation. A lazy evaluator simulates this duplication by inserting two pointers pointing to the actual argument. Sharing is a natural companion for non-strict evaluation as it avoids re-evaluation of expressions that are duplicated before they are evaluated.

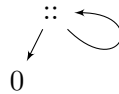
The underlying formalism that is typically used to obtain sharing for functional programming languages is term graph rewriting [30, 31]. Term graph rewriting [11, 32] uses graphs to represent terms thus allowing multiple arcs to point to the same node. For example, term graphs allows us to change the representation of the term rewrite rule defining the function *from* by replacing



which shares the variable x by having two arcs pointing to it.

While infinitary term rewriting is used to model the non-strictness of lazy evaluation, term graph rewriting models the sharing part of it. By endowing term graph rewriting with a notion of convergence, we aim to unify the two formalisms into one calculus, thus allowing us to model both aspects withing the same calculus.

1.1.2. Rational Terms. Term graphs can do more than only share common subexpressions. Through cycles term graphs may also provide a finite representation of certain infinite terms – so-called *rational terms*. For example, the infinite term $0 :: 0 :: 0 :: \dots$ can be represented as the finite term graph



Since a single node on a cycle in a term graph represents infinitely many corresponding subterms, the contraction of a single term graph redex may correspond to a transfinite term reduction that contracts infinitely many term redexes. For example, if we apply the rewrite rule $0 \rightarrow s(0)$ to the above term graph, we obtain a term graph that represents the term $s(0) :: s(0) :: s(0) :: \dots$, which can only be obtained from the term $0 :: 0 :: 0 :: \dots$ via a *transfinite* term reduction with the rule $0 \rightarrow s(0)$. Kennaway et al. [26] investigated this correspondence between cyclic term graph rewriting and infinitary term rewriting. Among other results they characterise a subset of transfinite term reductions – called *rational reductions* – that can be simulated by a corresponding finite term graph reduction. The above reduction from the term $0 :: 0 :: 0 :: \dots$ is an example of such a rational reduction.

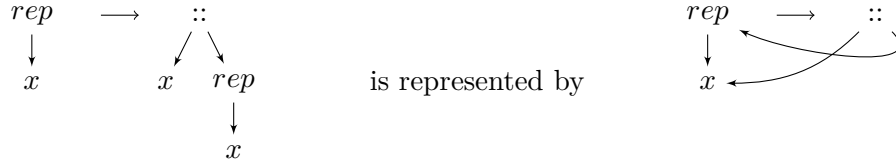
With the help of a unified formalism for infinitary and term graph rewriting, it should be easier to study the correspondence between infinitary term rewriting and finitary term graph rewriting further. The move from an infinitary term rewriting system to a term graph rewriting system only amounts to a change in the degree of sharing if we use infinitary term graph rewriting as a common framework.

Reconsider the term rewrite rule $\text{rep}(x) \rightarrow x :: \text{rep}(x)$, which defines a function rep that repeats its argument infinitely often:

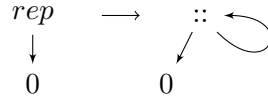
$$\text{rep}(0) \rightarrow 0 :: \text{rep}(0) \rightarrow 0 :: 0 :: \text{rep}(0) \rightarrow 0 :: 0 :: 0 :: \text{rep}(0) \rightarrow \dots \quad 0 :: 0 :: 0 :: \dots$$

This reduction happens to be not a rational reduction in the sense of Kennaway et al. [26].

The move from the term rule $\text{rep}(x) \rightarrow x :: \text{rep}(x)$ to a term graph rule is a simple matter of introducing sharing of common subexpressions:



Instead of creating a fresh copy of the redex on the right-hand side, the redex is reused by placing an edge from the right-hand side of the rule to its left-hand side. This allows us to represent the infinite reduction approximating the infinite term $0 :: 0 :: 0 :: \dots$ with the following single step term graph reduction induced by the above term graph rule:



Via its cyclic structure the resulting term graph represents the infinite term $0 :: 0 :: 0 :: \dots$.

Since both transfinite term reductions and the corresponding finite term graph reductions can be treated within the same formalism, we hope to provide a tool for studying the ability of cyclic term graph rewriting to finitely represent transfinite term reductions.

1.2. Contributions & Related Work.

1.2.1. *Contributions.* The main contributions of this paper are the following:

- (i) We devise a partial order on term graphs based on a restricted class of graph homomorphisms. We show that this partial order forms a complete semilattice and thus is technically suitable for defining a notion of convergence (Theorem 5.15). Moreover, we

illustrate alternative partial orders and show why they are not suitable for formalising convergence on term graphs.

- (ii) Independently, we devise a metric on term graphs and show that it forms a complete ultrametric space on term graphs (Theorem 7.4).
- (iii) Based on the partial order respectively the metric we define a notion of *weak convergence* for infinitary term graph rewriting. We show that – similar to the term rewriting case [7] – the metric calculus of infinitary term graph rewriting is the *total fragment* of the partial order calculus of infinitary term graph rewriting (Theorem 8.10).
- (iv) We confirm that the partial order and the metric on term graphs generalise the partial order respectively the metric that is used for infinitary term rewriting (Proposition 5.19 and 6.16). Moreover, we show that the corresponding notions of convergence are preserved by unravelling term graphs to terms thus establishing the soundness of our notions of convergence on term graphs w.r.t. the convergence on terms (Theorems 9.9 and 9.11).
- (v) We substantiate the appropriateness of our calculi by a number of examples that illustrate how increasing the sharing gradually reduces the number of steps necessary to reach the final result – eventually, from an infinite number of steps to a finite number (Sections 8 and 9).

1.2.2. *Related Work.* Calculi with explicit sharing and/or recursion, e.g. via *letrec*, can also be considered as a form of term graph rewriting. Ariola and Klop [3] recognised that adding such an explicit recursion mechanism to the lambda calculus may break confluence. In order to reconcile this, Ariola and Blom [2, 1] developed a notion of skew confluence that allows them to define an infinite normal form in the vein of Böhm trees.

Recently, we have investigated other notions of convergence for term graph rewriting [10, 9] that use simpler variants of the partial order and the metric that we use in this paper. Both of them have theoretically pleasing properties, e.g. the ideal completion and the metric completion of the set of finite term graphs both yield the set of all term graphs. However, the resulting notions of weak convergence are not fully satisfying and in fact counterintuitive for some cases. We will discuss this alternative approach and compare it to the present approach in more detail in Sections 5 and 6.

1.3. **Overview.** The structure of this paper is as follows: in Section 2, we provide the necessary background for metric spaces, partially ordered sets and term rewriting. In Section 3, we give an overview of infinitary term rewriting. Section 4 provides the necessary theory for graphs and term graphs. Sections 5 and 6 form the core of this paper. In these sections we study the partial order and the metric on term graphs that are the basis for the modes of convergence we propose in this paper. In Section 7, we then compare the two resulting modes of convergence. Moreover, in Section 8, we use these two modes of convergence to study two corresponding infinitary term graph rewriting calculi. In Section 9, we study the correspondences between infinitary term graph rewriting and infinitary term rewriting.

Some proofs have been omitted from the main body of the text. These proofs can be found in the appendix of this paper.

CONTENTS

1. Introduction	1
1.1. Motivation	2
1.2. Contributions & Related Work	4
1.3. Overview	5
2. Preliminaries	6
2.1. Sequences	7
2.2. Metric Spaces	7
2.3. Partial Orders	7
2.4. Terms	8
2.5. Term Rewriting Systems	8
3. Infinitary Term Rewriting	9
4. Graphs & Term Graphs	11
4.1. Homomorphisms	13
4.2. Isomorphisms & Isomorphism Classes	15
5. A Rigid Partial Order on Term Graphs	19
5.1. Partial Orders on Term Graphs	20
5.2. The Rigid Partial Order	24
6. A Rigid Metric on Term Graphs	30
6.1. Truncating Term Graphs	31
6.2. The Effect of Truncation	33
6.3. Deriving a Metric on Term Graphs	36
7. Metric vs. Partial Order Convergence	37
8. Infinitary Term Graph Rewriting	40
8.1. Term Graph Rewriting Systems	40
8.2. Convergence of Transfinite Reductions	42
9. Term Graph Rewriting vs. Term Rewriting	46
9.1. Soundness & Completeness of Infinitary Term Graph Rewriting	46
9.2. Preservation of Convergence under Unravelling	47
9.3. Finite Representations of Transfinite Term Reductions	50
10. Conclusions & Future Work	52
Acknowledgement	53
References	53
Appendix A. Proof of Lemma 5.14	54
Appendix B. Proof of Lemma 6.10	57

2. PRELIMINARIES

We assume the reader to be familiar with the basic theory of ordinal numbers, orders and topological spaces [21], as well as term rewriting [34]. In order to make this paper self-contained, we briefly recall all necessary preliminaries.

2.1. Sequences. We use the von Neumann definition of ordinal numbers. That is, an *ordinal number* (or simply *ordinal*) α is the set of all ordinal numbers strictly smaller than α . In particular, each natural number $n \in \mathbb{N}$ is an ordinal number with $n = \{0, 1, \dots, n-1\}$. The least infinite ordinal number is denoted by ω and is the set of all natural numbers. Ordinal numbers will be denoted by lower case Greek letters $\alpha, \beta, \gamma, \lambda, \iota$.

A *sequence* S of *length* α in a set A , written $(a_\iota)_{\iota < \alpha}$, is a function from α to A with $\iota \mapsto a_\iota$ for all $\iota \in \alpha$. We use $|S|$ to denote the length α of S . If α is a limit ordinal, then S is called *open*. Otherwise, it is called *closed*. If α is a finite ordinal, then S is called *finite*. Otherwise, it is called *infinite*. For a finite sequence $(a_i)_{i < n}$ we also use the notation $\langle a_0, a_1, \dots, a_{n-1} \rangle$. In particular, $\langle \rangle$ denotes the empty sequence. We write A^* for the set of all finite sequences in A .

The *concatenation* $(a_\iota)_{\iota < \alpha} \cdot (b_\iota)_{\iota < \beta}$ of two sequences is the sequence $(c_\iota)_{\iota < \alpha + \beta}$ with $c_\iota = a_\iota$ for $\iota < \alpha$ and $c_{\alpha + \iota} = b_\iota$ for $\iota < \beta$. A sequence S is a (proper) *prefix* of a sequence T , denoted $S \leq T$ (respectively $S < T$), if there is a (non-empty) sequence S' with $S \cdot S' = T$. The prefix of T of length $\beta \leq |T|$ is denoted $T|_\beta$. Similarly, a sequence S is a (proper) *suffix* of a sequence T if there is a (non-empty) sequence S' with $S' \cdot S = T$.

2.2. Metric Spaces. A pair (M, \mathbf{d}) is called a *metric space* if \mathbf{d} is a *metric* on the set M . That is, $\mathbf{d}: M \times M \rightarrow \mathbb{R}_0^+$ is a function satisfying $\mathbf{d}(x, y) = 0$ iff $x = y$ (identity), $\mathbf{d}(x, y) = \mathbf{d}(y, x)$ (symmetry), and $\mathbf{d}(x, z) \leq \mathbf{d}(x, y) + \mathbf{d}(y, z)$ (triangle inequality), for all $x, y, z \in M$. If \mathbf{d} instead of the triangle inequality, satisfies the stronger property $\mathbf{d}(x, z) \leq \max\{\mathbf{d}(x, y), \mathbf{d}(y, z)\}$ (strong triangle), then (M, \mathbf{d}) is called an *ultrametric space*.

Let $(a_\iota)_{\iota < \alpha}$ be a sequence in a metric space (M, \mathbf{d}) . The sequence $(a_\iota)_{\iota < \alpha}$ *converges* to an element $a \in M$, written $\lim_{\iota \rightarrow \alpha} a_\iota$, if, for each $\varepsilon \in \mathbb{R}^+$, there is a $\beta < \alpha$ such that $\mathbf{d}(a, a_\iota) < \varepsilon$ for every $\beta < \iota < \alpha$; $(a_\iota)_{\iota < \alpha}$ is *continuous* if $\lim_{\iota \rightarrow \lambda} a_\iota = a_\lambda$ for each limit ordinal $\lambda < \alpha$. Intuitively speaking, $(a_\iota)_{\iota < \alpha}$ converges to a if the metric distance between the elements a_ι of the sequence and a tends to 0 as the index ι approaches α , i.e. they approximate a arbitrarily well. Accordingly, $(a_\iota)_{\iota < \alpha}$ is continuous if it does not leap to a distant object at limit ordinal indices.

The sequence $(a_\iota)_{\iota < \alpha}$ is called *Cauchy* if, for any $\varepsilon \in \mathbb{R}^+$, there is a $\beta < \alpha$ such that, for all $\beta < \iota < \iota' < \alpha$, we have that $\mathbf{d}(a_\iota, a_{\iota'}) < \varepsilon$. That is, the elements a_ι of the sequence move closer and closer to each other as the index ι approaches α .

A metric space is called *complete* if each of its non-empty Cauchy sequences converges. That is, whenever the elements a_ι of a sequence move closer and closer together, they in fact approximate an existing object of the metric space, viz. $\lim_{\iota \rightarrow \alpha} a_\iota$.

2.3. Partial Orders. A *partial order* \leq on a set A is a binary relation on A such that $x \leq y, y \leq z$ implies $x \leq z$ (transitivity); $x \leq x$ (reflexivity); and $x \leq y, y \leq x$ implies $x = y$ (antisymmetry) for all $x, y, z \in A$. The pair (A, \leq) is then called a *partially ordered set*. A subset D of the underlying set A is called *directed* if it is non-empty and each pair of elements in D has an upper bound in D . A partially ordered set (A, \leq) is called a *complete partial order* (cpo) if it has a least element and each directed set D has a *least upper bound* (lub) $\sqcup D$. A cpo (A, \leq) is called a *complete semilattice* if every non-empty set B has *greatest lower bound* (glb) $\sqcap B$. In particular, this means that, in a complete semilattice, the *limit inferior* of any sequence $(a_\iota)_{\iota < \alpha}$, defined by $\liminf_{\iota \rightarrow \alpha} a_\iota = \sqcup_{\beta < \alpha} \left(\sqcap_{\beta \leq \iota < \alpha} a_\iota \right)$, always exists.

There is also an alternative characterisation of complete semilattices: a partially ordered set (A, \leq) is called *bounded complete* if each set $B \subseteq A$ that has an upper bound in A also has a least upper bound in A . Two elements $a, b \in A$ are called *compatible* if they have a common upper bound, i.e. there is some $c \in A$ with $a, b \leq c$.

Proposition 2.1 (bounded complete cpo = complete semilattice, [19]). *Given a cpo (A, \leq) the following are equivalent:*

- (i) (A, \leq) is a complete semilattice.
- (ii) (A, \leq) is bounded complete.
- (iii) Each two compatible elements in A have a least upper bound. □

Given two partially ordered sets (A, \leq_A) and (B, \leq_B) , a function $\phi: A \rightarrow B$ is called *monotonic* iff $a_1 \leq_A a_2$ implies $\phi(a_1) \leq_B \phi(a_2)$. In particular, we have that a sequence $(b_\iota)_{\iota < \alpha}$ in (B, \leq_B) is monotonic if $b_\iota \leq_B b_\gamma$ for all $\iota \leq \gamma < \alpha$.

2.4. Terms. Since we are interested in the infinitary calculus of term rewriting, we consider the set $\mathcal{T}^\infty(\Sigma)$ of *infinitary terms* (or simply *terms*) over some *signature* Σ . A *signature* Σ is a countable set of symbols such that each symbol $f \in \Sigma$ is associated with an arity $\text{ar}(f) \in \mathbb{N}$, and we write $\Sigma^{(n)}$ for the set of symbols in Σ that have arity n . The set $\mathcal{T}^\infty(\Sigma)$ is defined as the *greatest* set T such that $t \in T$ implies $t = f(t_1, \dots, t_k)$ for some $f \in \Sigma^{(k)}$ and $t_1, \dots, t_k \in T$. For each constant symbol $c \in \Sigma^{(0)}$, we write c for the term $c()$. For a term $t \in \mathcal{T}^\infty(\Sigma)$ we use the notation $\mathcal{P}(t)$ to denote the *set of positions* in t . $\mathcal{P}(t)$ is the least subset of \mathbb{N}^* such that $\langle \rangle \in \mathcal{P}(t)$ and $\langle i \rangle \cdot \pi \in \mathcal{P}(t)$ if $t = f(t_0, \dots, t_{k-1})$ with $0 \leq i < k$ and $\pi \in \mathcal{P}(t_i)$. For terms $s, t \in \mathcal{T}^\infty(\Sigma)$ and a position $\pi \in \mathcal{P}(t)$, we write $t|_\pi$ for the *subterm* of t at π , $t(\pi)$ for the function symbol in t at π , and $t[s]_\pi$ for the term t with the subterm at π replaced by s . As positions are sequences, we use the prefix order \leq defined on them. A position is also called an *occurrence* if the focus lies on the subterm at that position rather than the position itself. The set $\mathcal{T}(\Sigma)$ of *finite terms* is the subset of $\mathcal{T}^\infty(\Sigma)$ that contains all terms with a finite set of positions.

On $\mathcal{T}^\infty(\Sigma)$ a similarity measure $\text{sim}(\cdot, \cdot): \mathcal{T}^\infty(\Sigma) \times \mathcal{T}^\infty(\Sigma) \rightarrow \omega + 1$ is defined as follows

$$\text{sim}(s, t) = \min \{ |\pi| \mid \pi \in \mathcal{P}(s) \cap \mathcal{P}(t), s(\pi) \neq t(\pi) \} \cup \{ \omega \} \quad \text{for } s, t \in \mathcal{T}^\infty(\Sigma)$$

That is, $\text{sim}(s, t)$ is the minimal depth at which s and t differ, respectively ω if $s = t$. Based on this similarity measure, a distance function \mathbf{d} is defined by $\mathbf{d}(s, t) = 2^{-\text{sim}(s, t)}$, where we interpret $2^{-\omega}$ as 0. The pair $(\mathcal{T}^\infty(\Sigma), \mathbf{d})$ is known to form a complete ultrametric space [4].

Partial terms, i.e. terms over signature $\Sigma_\perp = \Sigma \uplus \{\perp\}$ with \perp a fresh nullary symbol, can be endowed with a binary relation \leq_\perp by defining $s \leq_\perp t$ iff s can be obtained from t by replacing some subterm occurrences in t by \perp . Interpreting the term \perp as denoting “undefined”, \leq_\perp can be read as “is less defined than”. The pair $(\mathcal{T}^\infty(\Sigma_\perp), \leq_\perp)$ is known to form a complete semilattice [16]. To explicitly distinguish them from partial terms, we call terms in $\mathcal{T}^\infty(\Sigma)$ *total*.

2.5. Term Rewriting Systems. For term rewriting systems, we have to consider terms with variables. To this end, we assume a countably infinite set \mathcal{V} of variables and extend a signature Σ to a signature $\Sigma_\mathcal{V} = \Sigma \uplus \mathcal{V}$ with variables in \mathcal{V} as nullary symbols. Instead of $\mathcal{T}^\infty(\Sigma_\mathcal{V})$ we also write $\mathcal{T}^\infty(\Sigma, \mathcal{V})$. A *term rewriting system* (TRS) \mathcal{R} is a pair (Σ, R) consisting of a signature Σ and a set R of *term rewrite rules* of the form $l \rightarrow r$ with

$l \in \mathcal{T}^\infty(\Sigma, \mathcal{V}) \setminus \mathcal{V}$ and $r \in \mathcal{T}^\infty(\Sigma, \mathcal{V})$ such that all variables occurring in r also occur in l . Note that both the left- and the right-hand side may be infinite. We usually use x, y, z and primed respectively indexed variants thereof to denote variables in \mathcal{V} . A *substitution* σ is a mapping from \mathcal{V} to $\mathcal{T}^\infty(\Sigma, \mathcal{V})$. Such a substitution σ can be uniquely lifted to a homomorphism from $\mathcal{T}^\infty(\Sigma, \mathcal{V})$ to $\mathcal{T}^\infty(\Sigma, \mathcal{V})$ mapping a term $t \in \mathcal{T}^\infty(\Sigma, \mathcal{V})$ to $t\sigma$ by setting $x\sigma = \sigma(x)$ if $x \in \mathcal{V}$ and $f(t_1, \dots, t_n)\sigma = f(t_1\sigma, \dots, t_n\sigma)$ if $f \in \Sigma^{(n)}$.

As in the finitary setting, every TRS \mathcal{R} defines a *rewrite relation* $\rightarrow_{\mathcal{R}}$ that indicates *rewrite steps*:

$$s \rightarrow_{\mathcal{R}} t \iff \exists \pi \in \mathcal{P}(s), l \rightarrow r \in R, \sigma: s|_{\pi} = l\sigma, t = s[r\sigma]_{\pi}$$

Instead of $s \rightarrow_{\mathcal{R}} t$, we sometimes write $s \rightarrow_{\pi, \rho} t$ in order to indicate the applied rule ρ and the position π , or simply $s \rightarrow t$. The subterm $s|_{\pi}$ is called a ρ -*redex* or simply *redex*, $r\sigma$ its *contractum*, and $s|_{\pi}$ is said to be *contracted* to $r\sigma$.

3. INFINITARY TERM REWRITING

Before pondering over the right approach to an infinitary calculus of term graph rewriting, we want to provide a brief overview of infinitary term rewriting [24, 7, 13]. This should give an insight into the different approaches to dealing with infinite reductions. However, in contrast to the majority of the literature on infinitary term rewriting, which is concerned with strong convergence [24, 27], we will only consider weak notions of convergence in this paper; cf. [14, 20, 33]. This weak form of convergence, also called *Cauchy convergence*, is entirely based on the sequence of objects produced by rewriting without considering *how* the rewrite rules are applied.

A (*transfinite*) *reduction* in a term rewriting system \mathcal{R} , is a sequence $S = (t_i \rightarrow_{\mathcal{R}} t_{i+1})_{i < \hat{\alpha}}$ of rewrite steps in \mathcal{R} . Note that the underlying sequence of terms $(t_i)_{i < \hat{\alpha}}$ has length $\hat{\alpha}$, where $\hat{\alpha} = \alpha$ if S is open, and $\hat{\alpha} = \alpha + 1$ if S is closed. The reduction S is called *m-continuous* in \mathcal{R} , written $S: t_0 \xrightarrow{m}_{\mathcal{R}} \dots$, if the sequence of terms $(t_i)_{i < \hat{\alpha}}$ is continuous in $(\mathcal{T}^\infty(\Sigma), \mathbf{d})$, i.e. $\lim_{i \rightarrow \lambda} t_i = t_\lambda$ for each limit ordinal $\lambda < \alpha$. The reduction S is said to *m-converge* to a term t in \mathcal{R} , written $S: t_0 \xrightarrow{m}_{\mathcal{R}} t$, if it is *m-continuous* and $\lim_{i \rightarrow \hat{\alpha}} t_i = t$.

Example 3.1. Consider the term rewriting system \mathcal{R} containing the rule $\rho_1: a::x \rightarrow b::a::x$. By repeatedly applying ρ_1 , we obtain the infinite reduction

$$S: a::c \rightarrow b::a::c \rightarrow b::b::a::c \rightarrow \dots$$

The position at which two consecutive terms differ moves deeper and deeper during the reduction S , i.e. the \mathbf{d} -distance between them tends to 0. Hence, S *m-converges* to the infinite term $s = b::b::b::\dots$, i.e. $S: a::c \xrightarrow{m} s$.

Now consider a TRS with the slightly different rule $\rho_2: a::x \rightarrow a::b::x$. This TRS yields a reduction

$$S': a::c \rightarrow a::b::c \rightarrow a::b::b::c \rightarrow \dots$$

Even though the rule ρ_2 is applied at the root of the term in each step of S' , the \mathbf{d} -distance between two consecutive terms tends to 0 again. The reduction S' *m-converges* to the infinite term $s' = a::b::b::\dots$, i.e. $S': a::c \xrightarrow{m} s'$.

In contrast to the weak *m-convergence* that we consider here, strong *m-convergence* [24, 27] additionally requires that the depth of the contracted redexes tends to infinity as the reduction approaches a limit ordinal. Concerning Example 3.1 above, we have for instance

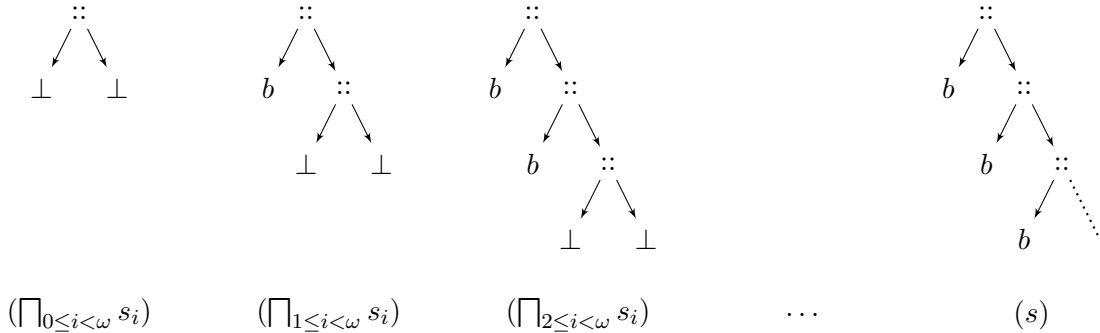
that S also strongly m -converges – the rule is applied at increasingly deep redexes – whereas S' does not strongly m -converge – each step in S' results from a contraction at the root.

In the partial order model of infinitary rewriting [7], convergence is defined via the limit inferior in the complete semilattice $(\mathcal{T}^\infty(\Sigma_\perp), \leq_\perp)$. Given a TRS $\mathcal{R} = (\Sigma, R)$, we extend it to $\mathcal{R}_\perp = (\Sigma_\perp, R)$ by adding the fresh constant symbol \perp such that it admits all terms in $\mathcal{T}^\infty(\Sigma_\perp)$. A reduction $S = (t_\iota \rightarrow_{\mathcal{R}_\perp} t_{\iota+1})_{\iota < \alpha}$ in this system \mathcal{R}_\perp is called p -continuous in \mathcal{R} , written $S: t_0 \xrightarrow{\mathcal{R}} \dots$, if $\liminf_{\iota \rightarrow \lambda} t_\iota = t_\lambda$ for each limit ordinal $\lambda < \alpha$. The reduction S is said to p -converge to a term t in \mathcal{R} , written $S: t_0 \xrightarrow{\mathcal{R}} t$, if it is p -continuous and $\liminf_{\iota \rightarrow \hat{\alpha}} t_\iota = t$.

The distinguishing feature of the partial order approach is that each continuous reduction also converges due to the semilattice structure of partial terms. Moreover, p -convergence provides a conservative extension to m -convergence that allows rewriting modulo *meaningless terms* [7] by essentially mapping those parts of the reduction to \perp that are divergent according to the metric mode of convergence.

Intuitively, the limit inferior in $(\mathcal{T}^\infty(\Sigma_\perp), \leq_\perp)$ – and thus p -convergence – describes an approximation process that accumulates each piece of information that remains *stable* from some point onwards. This is based on the ability of the partial order \leq_\perp to capture a notion of *information preservation*, i.e. $s \leq_\perp t$ iff t contains at least the same information as s does but potentially more. A monotonic sequence of terms $t_0 \leq_\perp t_1 \leq_\perp \dots$ thus approximates the information contained in $\bigsqcup_{i < \omega} t_i$. Given this reading of \leq_\perp , the glb $\prod T$ of a set of terms T captures the common (non-contradicting) information of the terms in T . Leveraging this observation, a sequence that is not necessarily monotonic can be turned into a monotonic sequence $t_j = \prod_{j \leq i < \omega} s_i$ such that each t_j contains exactly the information that remains stable in $(s_i)_{i < \omega}$ from j onwards. Hence, the limit inferior $\liminf_{i \rightarrow \omega} s_i = \bigsqcup_{j < \omega} \prod_{j \leq i < \omega} s_i$ is the term that contains the accumulated information that eventually remains stable in $(s_i)_{i < \omega}$. This is expressed as an approximation of the monotonically increasing information that remains stable from some point on.

Example 3.2. Reconsider the system from Example 3.1. The reduction S also p -converges to s . This can be seen by forming the sequence $(\prod_{j \leq i < \omega} s_i)_{i < \omega}$ of stable information of the underlying sequence $(s_i)_{i < \omega}$ of terms in S :

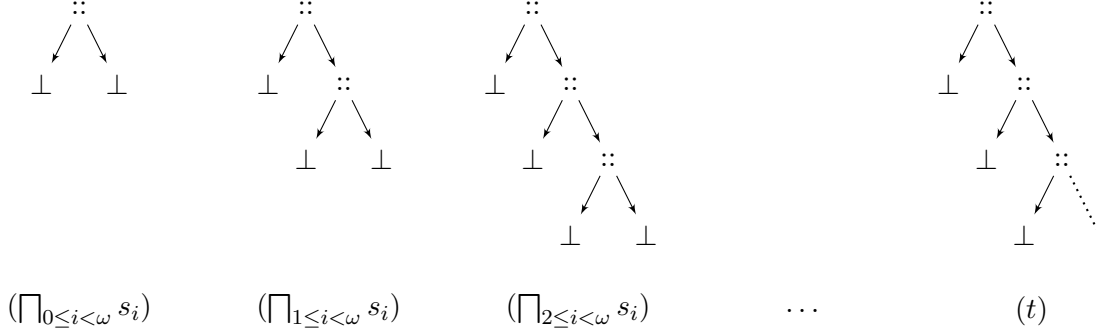


This sequence approximates the term $s = b :: b :: b :: \dots$.

Now consider the rule ρ_1 together with the rule $\rho_3: b :: x \rightarrow a :: b :: x$. Starting with the same term, but applying the two rules alternately at the root, we obtain the reduction sequence

$$T: a :: c \rightarrow b :: a :: c \rightarrow a :: b :: a :: c \rightarrow b :: a :: b :: a :: c \rightarrow \dots$$

Now the differences between two consecutive terms occur right below the root symbol “ $::$ ”. Hence, T does not m -converge. This, however, only affects the left argument of each “ $::$ ”. Following the right argument position, the bare list structure becomes eventually stable. The sequence $(\prod_{j \leq i < \omega} s_i)_{i < \omega}$ of stable information



approximates the term $t = \perp :: \perp :: \perp \dots$. Hence, T p -converges to t .

Note that in both the metric and the partial order setting continuity is simply the convergence of every proper prefix: a reduction $S = (t_i \rightarrow t_{i+1})_{i < \alpha}$ is m -continuous (respectively p -continuous) iff every proper prefix $S|_\beta$ m -converges (respectively p -converges) to t_β .

In order to define p -convergence, we had to extend terms with partiality. However, apart from this extension, both m - and p -convergence coincide. To describe this more precisely we use the following terms: a reduction $S: s \rightarrow \dots$ is *p -continuous in $\mathcal{T}^\infty(\Sigma)$* iff each term in S is total, i.e. in $\mathcal{T}^\infty(\Sigma)$; a reduction $S: s \rightarrow t$ is called *p -convergent in $\mathcal{T}^\infty(\Sigma)$* iff t and each term in S is total. We then have the following theorem:

Theorem 3.3 (p -convergence in $\mathcal{T}^\infty(\Sigma) = m$ -convergence, [5]). *For every reduction S in a TRS the following equivalences hold:*

- (i) $S: s \rightarrow t$ in $\mathcal{T}^\infty(\Sigma)$ iff $S: s \rightarrow t$
- (ii) $S: s \rightarrow \dots$ in $\mathcal{T}^\infty(\Sigma)$ iff $S: s \rightarrow \dots$ □

Example 3.2 illustrates the correspondence between p - and m -convergence: the reduction S p -converges in $\mathcal{T}^\infty(\Sigma)$ and m -converges whereas the reduction T p -converges but not in $\mathcal{T}^\infty(\Sigma)$ and thus does not m -converge.

Kennaway [22] and Bahr [6] investigated abstract models of infinitary rewriting based on metric spaces respectively partially ordered sets. We shall take these abstract models as a basis to formulate a theory of infinitary term graph reductions. The key question that we have to address is what an appropriate metric space respectively partial order on term graphs looks like.

4. GRAPHS & TERM GRAPHS

This section provides the basic notions for term graphs and more generally for graphs. Terms over a signature, say Σ , can be thought of as rooted trees whose nodes are labelled with symbols from Σ . Moreover, in these trees a node labelled with a k -ary symbol is restricted to have out-degree k and the outgoing edges are ordered. In this way the i -th successor of a node labelled with a symbol f is interpreted as the root node of the subtree

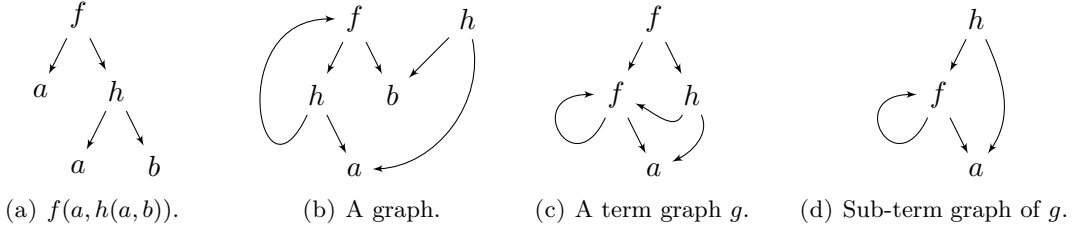


FIGURE 1. Tree representation of a term and generalisation to (term) graphs.

that represents the i -th argument of f . For example, consider the term $f(a, h(a, b))$. The corresponding representation as a tree is shown in Figure 1a.

In term graphs, the restriction to a tree structure is abolished. The corresponding notion of term graphs we are using is taken from Barendregt et al. [11]. We begin by defining the underlying notion of graphs.

Definition 4.1 (graphs). Let Σ be a signature. A *graph* over Σ is a tuple $g = (N, \text{lab}, \text{suc})$ consisting of a set N (of *nodes*), a *labelling function* $\text{lab}: N \rightarrow \Sigma$, and a *successor function* $\text{suc}: N \rightarrow N^*$ such that $|\text{suc}(n)| = \text{ar}(\text{lab}(n))$ for each node $n \in N$, i.e. a node labelled with a k -ary symbol has precisely k successors. The graph g is called *finite* whenever the underlying set N of nodes is finite. If $\text{suc}(n) = \langle n_0, \dots, n_{k-1} \rangle$, then we write $\text{suc}_i(n)$ for n_i . Moreover, we use the abbreviation $\text{ar}_g(n)$ for the arity $\text{ar}(\text{lab}(n))$ of n .

Example 4.2. Let $\Sigma = \{f/2, h/2, a/0, b/0\}$ be a signature. The graph over Σ , depicted in Figure 1b, is given by the triple $(N, \text{lab}, \text{suc})$ with $N = \{n_0, n_1, n_2, n_3, n_4\}$, $\text{lab}(n_0) = f$, $\text{lab}(n_1) = \text{lab}(n_4) = h$, $\text{lab}(n_2) = b$, $\text{lab}(n_3) = a$ and $\text{suc}(n_0) = \langle n_1, n_2 \rangle$, $\text{suc}(n_1) = \langle n_0, n_3 \rangle$, $\text{suc}(n_2) = \text{suc}(n_3) = \langle \rangle$, $\text{suc}(n_4) = \langle n_2, n_3 \rangle$.

Definition 4.3 (paths, reachability). Let $g = (N, \text{lab}, \text{suc})$ be a graph and $n, m \in N$.

- (i) A *path* in g from n to m is a finite sequence $\pi \in \mathbb{N}^*$ such that either
 - π is empty and $n = m$, or
 - $\pi = \langle i \rangle \cdot \pi'$ with $0 \leq i < \text{ar}_g(n)$ and the suffix π' is a path in g from $\text{suc}_i(n)$ to m .
- (ii) If there exists a path from n to m in g , we say that m is *reachable* from n in g .

Since paths are sequences, we may use the prefix order on sequences for paths as well. That is, we write $\pi_1 \leq \pi_2$ (respectively $\pi_1 < \pi_2$) if there is a (non-empty) path π_3 with $\pi_1 \cdot \pi_3 = \pi_2$.

Definition 4.4 (term graphs). Given a signature Σ , a *term graph* g over Σ is a tuple $(N, \text{lab}, \text{suc}, r)$ consisting of an *underlying graph* $(N, \text{lab}, \text{suc})$ over Σ whose nodes are all reachable from the *root node* $r \in N$. The term graph g is called *finite* if the underlying graph is finite, i.e. the set N of nodes is finite. The class of all term graphs over Σ is denoted $\mathcal{G}^\infty(\Sigma)$; the class of all finite term graphs over Σ is denoted $\mathcal{G}(\Sigma)$. We use the notation N^g , lab^g , suc^g and r^g to refer to the respective components $N, \text{lab}, \text{suc}$ and r of g . In analogy to subterms, term graphs have *sub-term graphs*. Given a graph or a term graph h and a node n in h , we write $h|_n$ to denote the sub-term graph of h rooted in n .

Example 4.5. Let $\Sigma = \{f/2, h/2, c/0\}$ be a signature. The term graph g over Σ , depicted in Figure 1c, is given by the quadruple $(N, \text{lab}, \text{suc}, r)$, where $N = \{r, n_1, n_2, n_3\}$, $\text{suc}(r) = \langle n_1, n_2 \rangle$, $\text{suc}(n_1) = \langle n_1, n_3 \rangle$, $\text{suc}(n_2) = \langle n_1, n_3 \rangle$, $\text{suc}(n_3) = \langle \rangle$ and $\text{lab}(r) = \text{lab}(n_1) = f$, $\text{lab}(n_2) = h$, $\text{lab}(n_3) = c$. Figure 1d depicts the sub-term graph $g|_{n_2}$ of g .

Paths in a graph are not absolute but relative to a starting node. In term graphs, however, we have a distinguished root node from which each node is reachable. Paths relative to the root node are central for dealing with term graphs:

Definition 4.6 (positions, depth, cyclicity, trees). Let $g \in \mathcal{G}^\infty(\Sigma)$ and $n \in N^g$.

- (i) A *position* of n in g is a path in the underlying graph of g from r^g to n . The set of all positions in g is denoted $\mathcal{P}(g)$; the set of all positions of n in g is denoted $\mathcal{P}_g(n)$.¹
- (ii) The *depth* of n in g , denoted $\text{depth}_g(n)$, is the minimum of the lengths of the positions of n in g , i.e. $\text{depth}_g(n) = \min \{|\pi| \mid \pi \in \mathcal{P}_g(n)\}$.
- (iii) For a position $\pi \in \mathcal{P}(g)$, we write $\text{node}_g(\pi)$ for the unique node $n \in N^g$ with $\pi \in \mathcal{P}_g(n)$ and $g(\pi)$ for its symbol $\text{lab}^g(n)$.
- (iv) A position $\pi \in \mathcal{P}(g)$ is called *cyclic* if there are paths $\pi_1 < \pi_2 \leq \pi$ with $\text{node}_g(\pi_1) = \text{node}_g(\pi_2)$, i.e. π passes a node twice. The non-empty path π' with $\pi_1 \cdot \pi' = \pi_2$ is then called a *cycle* of $\text{node}_g(\pi_1)$. A position that is not cyclic is called *acyclic*. If g has a cyclic position, g is called cyclic; otherwise g is called acyclic.
- (v) The term graph g is called a *term tree* if each node in g has exactly one position.

Note that the labelling function of graphs – and thus term graphs – is *total*. In contrast, Barendregt et al. [11] considered *open* (term) graphs with a *partial* labelling function such that unlabelled nodes denote holes or variables. This is reflected in their notion of homomorphisms in which the homomorphism condition is suspended for unlabelled nodes.

4.1. Homomorphisms. Instead of a partial node labelling function for term graphs, we chose a *syntactic* approach that is closer to the representation in terms: variables, holes and “bottoms” are represented as distinguished syntactic entities. We achieve this on term graphs by making the notion of homomorphisms dependent on a set of constant symbols Δ for which the homomorphism condition is suspended:

Definition 4.7 (Δ -homomorphisms). Let Σ be a signature, $\Delta \subseteq \Sigma^{(0)}$, and $g, h \in \mathcal{G}^\infty(\Sigma)$.

- (i) A function $\phi: N^g \rightarrow N^h$ is called *homomorphic* in $n \in N^g$ if the following holds:

$$\text{lab}^g(n) = \text{lab}^h(\phi(n)) \quad (\text{labelling})$$

$$\phi(\text{succ}_i^g(n)) = \text{succ}_i^h(\phi(n)) \quad \text{for all } 0 \leq i < \text{ar}_g(n) \quad (\text{successor})$$

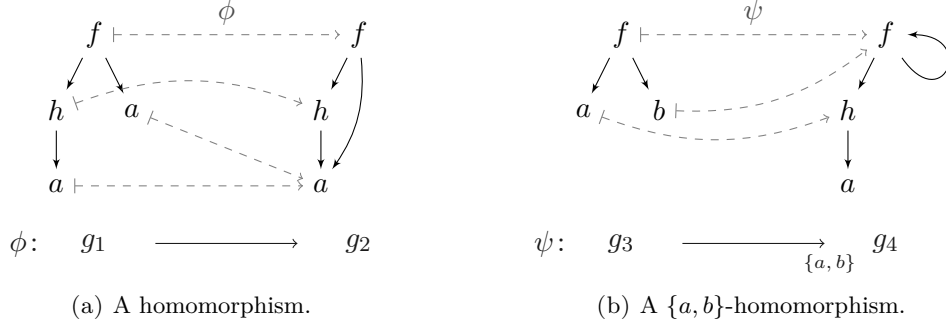
- (ii) A Δ -homomorphism ϕ from g to h , denoted $\phi: g \rightarrow_\Delta h$, is a function $\phi: N^g \rightarrow N^h$ that is homomorphic in n for all $n \in N^g$ with $\text{lab}^g(n) \notin \Delta$ and satisfies

$$\phi(r^g) = r^h \quad (\text{root})$$

Note that, for $\Delta = \emptyset$, we get the usual notion of homomorphisms on term graphs (e.g. Barendsen [12]). The Δ -nodes can be thought of as holes in the term graphs that can be filled with other term graphs. For example, if we have a distinguished set of variable symbols $\mathcal{V} \subseteq \Sigma^{(0)}$, we can use \mathcal{V} -homomorphisms to formalise the matching step of term graph rewriting, which requires the instantiation of variables.

Example 4.8. Figure 2 depicts two functions ϕ and ψ . Whereas ϕ is a homomorphism, the function ψ is not a homomorphism since, for example, the node labelled a in g_3 is mapped to a node labelled h in g_3 . Nevertheless, ψ is a $\{a, b\}$ -homomorphism. Note that

¹The notion/notation of positions is borrowed from terms: Every position π of a node n corresponds to the subterm represented by n occurring at position π in the unravelling of the term graph to a term.

FIGURE 2. Δ -homomorphisms.

Δ -homomorphisms may introduce additional sharing in the target term graph by mapping several nodes in the source to the same node in the target.

Proposition 4.9 (Δ -homomorphism preorder). *The Δ -homomorphisms on $\mathcal{G}^\infty(\Sigma)$ form a category that is a preorder, i.e. there is at most one Δ -homomorphism from one term graph to another.*

Proof. The identity Δ -homomorphism is obviously the identity mapping on the set of nodes. Moreover, an easy equational reasoning reveals that the composition of two Δ -homomorphisms is again a Δ -homomorphism. Associativity of this composition is obvious as Δ -homomorphisms are functions.

To show that the category is a preorder, assume that there are two Δ -homomorphisms $\phi_1, \phi_2: g \rightarrow_\Delta h$. We prove that $\phi_1 = \phi_2$ by showing that $\phi_1(n) = \phi_2(n)$ for all $n \in N^g$ by induction on the depth of n in g .

Let $\text{depth}_g(n) = 0$, i.e. $n = r^g$. By the root condition for ϕ , we have that $\phi_1(r^g) = r^h = \phi_2(r^g)$. Let $\text{depth}_g(n) = d > 0$. Then n has a position $\pi \cdot \langle i \rangle$ in g such that $\text{depth}_g(n') < d$ for $n' = \text{node}_g(\pi)$. Hence, we can employ the induction hypothesis for n' . Moreover, since n' has at least one successor node, viz. n , it cannot be labelled with a nullary symbol and a fortiori not with a symbol in Δ . Therefore, the Δ -homomorphisms ϕ_1 and ϕ_2 are homomorphic in n' and we can thus reason as follows:

$$\begin{aligned}
 \phi_1(n) &= \text{succ}_i^h(\phi_1(n')) && \text{(successor condition for } \phi_1) \\
 &= \text{succ}_i^h(\phi_2(n')) && \text{(ind. hyp.)} \\
 &= \phi_2(n) && \text{(successor condition for } \phi_2)
 \end{aligned}$$

□

As a consequence, whenever there are two Δ -homomorphisms $\phi: g \rightarrow_\Delta h$ and $\psi: h \rightarrow_\Delta g$, they are inverses of each other, i.e. Δ -isomorphisms. If two term graphs are Δ -isomorphic, we write $g \cong_\Delta h$.

For the two special cases $\Delta = \emptyset$ and $\Delta = \{\sigma\}$, we write $\phi: g \rightarrow h$ respectively $\phi: g \rightarrow_\sigma h$ instead of $\phi: g \rightarrow_\Delta h$ and call ϕ a *homomorphism* respectively a σ -*homomorphism*. The same convention applies to Δ -isomorphisms.

The structure of positions permits a convenient characterisation of Δ -homomorphisms:

Lemma 4.10 (characterisation of Δ -homomorphisms). *For $g, h \in \mathcal{G}^\infty(\Sigma)$, a function $\phi: N^g \rightarrow N^h$ is a Δ -homomorphism $\phi: g \rightarrow_\Delta h$ iff the following holds for all $n \in N^g$:*

$$(a) \mathcal{P}_g(n) \subseteq \mathcal{P}_h(\phi(n)), \quad \text{and} \quad (b) \text{lab}^g(n) \notin \Delta \implies \text{lab}^g(n) = \text{lab}^h(\phi(n)).$$

Proof. For the “only if” direction, assume that $\phi: g \rightarrow_\Delta h$. (b) is the labelling condition and is therefore satisfied by ϕ . To establish (a), we show the equivalent statement

$$\forall \pi \in \mathcal{P}(g). \forall n \in N^g. \pi \in \mathcal{P}_g(n) \implies \pi \in \mathcal{P}_h(\phi(n))$$

We do so by induction on the length of π : if $\pi = \langle \rangle$, then $\pi \in \mathcal{P}_g(n)$ implies $n = r^g$. By the root condition, we have $\phi(r^g) = r^h$ and, therefore, $\pi = \langle \rangle \in \phi(r^g)$. If $\pi = \pi' \cdot \langle i \rangle$, then let $n' = \text{node}_g(\pi')$. Consequently, $\pi' \in \mathcal{P}_g(n')$ and, by induction hypothesis, $\pi' \in \mathcal{P}_h(\phi(n'))$. Since $\pi = \pi' \cdot \langle i \rangle$, we have $\text{succ}_i^g(n') = n$. By the successor condition we can conclude $\phi(n) = \text{succ}_i^h(\phi(n'))$. This and $\pi' \in \mathcal{P}_h(\phi(n'))$ yields that $\pi' \cdot \langle i \rangle \in \mathcal{P}_h(\phi(n))$.

For the “if” direction, we assume (a) and (b). The labelling condition follows immediately from (b). For the root condition, observe that since $\langle \rangle \in \mathcal{P}_g(r^g)$, we also have $\langle \rangle \in \mathcal{P}_h(\phi(r^g))$. Hence, $\phi(r^g) = r^h$. In order to show the successor condition, let $n, n' \in N^g$ and $0 \leq i < \text{ar}_g(n)$ such that $\text{succ}_i^g(n) = n'$. Then there is a position $\pi \in \mathcal{P}_g(n)$ with $\pi \cdot \langle i \rangle \in \mathcal{P}_g(n')$. By (a), we can conclude that $\pi \in \mathcal{P}_h(\phi(n))$ and $\pi \cdot \langle i \rangle \in \mathcal{P}_h(\phi(n'))$ which implies that $\text{succ}_i^h(\phi(n)) = \phi(n')$. \square

By Proposition 4.9, there is at most one Δ -homomorphism between two term graphs. The lemma above uniquely defines this Δ -homomorphism: if there is a Δ -homomorphism from g to h , it is defined by $\phi(n) = n'$, where n' is the unique node $n' \in N^h$ with $\mathcal{P}_g(n) \subseteq \mathcal{P}_h(n')$. Moreover, while it is not true for arbitrary Δ -homomorphisms, we have that homomorphisms are surjective.

Lemma 4.11 (homomorphisms are surjective). *Every homomorphism $\phi: g \rightarrow h$, with $g, h \in \mathcal{G}^\infty(\Sigma)$, is surjective.*

Proof. Follows from an easy induction on the depth of the nodes in h . \square

The $\{a, b\}$ -homomorphism illustrated in Figure 2b, shows that the above lemma does not hold for Δ -homomorphisms in general.

4.2. Isomorphisms & Isomorphism Classes. When dealing with term graphs, in particular, when studying term graph transformations, we do not want to distinguish between isomorphic term graphs. Distinct but isomorphic term graphs do only differ in the naming of nodes and are thus an unwanted artifact of the definition of term graphs. In this way, equality up to isomorphism is similar to α -equivalence of λ -terms and has to be dealt with.

In this section, we characterise isomorphisms and more generally Δ -isomorphisms. From this we derive two canonical representations of isomorphism classes of term graphs. One is simply a subclass of the class of term graphs while the other one is based on the structure provided by the positions of term graphs. The relevance of the former representation is derived from the fact that we still have term graphs that can be easily manipulated whereas the latter is more technical and will be helpful for constructing term graphs up to isomorphism.

Note that a bijective Δ -homomorphism is not necessarily a Δ -isomorphism. To realise this, consider two term graphs g, h , each with one node only. Let the node in g be labelled with a and the node in h with b then the only possible a -homomorphism from g to h is

clearly a bijection but not an a -isomorphism. On the other hand, bijective homomorphisms indeed are isomorphisms.

Lemma 4.12 (bijective homomorphisms are isomorphisms). *Let $g, h \in \mathcal{G}^\infty(\Sigma)$ and $\phi: g \rightarrow h$. Then the following are equivalent*

- (a) ϕ is an isomorphism.
- (b) ϕ is bijective.
- (c) ϕ is injective.

Proof. The implication (a) \Rightarrow (b) is trivial. The equivalence (b) \Leftrightarrow (c) follows from Lemma 4.11. For the implication (b) \Rightarrow (a), consider the inverse ϕ^{-1} of ϕ . We need to show that ϕ^{-1} is a homomorphism from h to g . The root condition follows immediately from the root condition for ϕ . Similarly, an easy equational reasoning reveals that ϕ^{-1} is homomorphic in N^h since ϕ is homomorphic in all $n \in N^g$. \square

From the characterisation of Δ -homomorphisms in Lemma 4.10, we immediately obtain a characterisation of Δ -isomorphisms as follows:

Lemma 4.13 (characterisation of Δ -isomorphisms). *For all $g, h \in \mathcal{G}^\infty(\Sigma)$, a function $\phi: N^g \rightarrow N^h$ is a Δ -isomorphism iff for all $n \in N^g$*

- (a) $\mathcal{P}_h(\phi(n)) = \mathcal{P}_g(n)$, and
- (b) $\text{lab}^g(n) = \text{lab}^h(\phi(n))$ or $\text{lab}^g(n), \text{lab}^h(\phi(n)) \in \Delta$.

Proof. Immediate consequence of Lemma 4.10 and Proposition 4.9. \square

Note that whenever Δ is a singleton set, the condition $\text{lab}^g(n), \text{lab}^h(\phi(n)) \in \Delta$ in the above lemma implies $\text{lab}^g(n) = \text{lab}^h(\phi(n))$. Therefore, we obtain the following corollary:

Corollary 4.14 (σ -isomorphism = isomorphism). *Given $g, h \in \mathcal{G}^\infty(\Sigma)$ and $\sigma \in \Sigma^{(0)}$, we have $g \cong h$ iff $g \cong_\sigma h$.*

Note that the above equivalence does not hold for Δ -homomorphisms with more than one symbol in Δ : consider the term graphs $g = a$ and $h = b$ consisting of a single node labelled a respectively b . While g and h are Δ -isomorphic for $\Delta = \{a, b\}$, they are not isomorphic.

4.2.1. Canonical Term Graphs. From the Lemmas 4.12 and 4.13 we learned that isomorphisms between term graphs are bijections that preserve and reflect the positions as well as the labelling of each node. These findings motivate the following definition of canonical term graphs as candidates for representatives of isomorphism classes:

Definition 4.15 (canonical term graphs). A term graph g is called *canonical* if $n = \mathcal{P}_g(n)$ holds for each $n \in N^g$. That is, each node is the set of its positions in the term graph. The set of all (finite) canonical term graphs over Σ is denoted $\mathcal{G}_C^\infty(\Sigma)$ (respectively $\mathcal{G}_C(\Sigma)$). Given a term graph $h \in \mathcal{G}_C^\infty(\Sigma)$, its *canonical representative* $\mathcal{C}(h)$ is the canonical term graph given by

$$\begin{aligned} N^{\mathcal{C}(h)} &= \{\mathcal{P}_h(n) \mid n \in N\} & r^{\mathcal{C}(h)} &= \mathcal{P}_h(r) & \text{lab}^{\mathcal{C}(h)}(\mathcal{P}_h(n)) &= \text{lab}^h(n) \quad \text{for all } n \in N \\ \text{suc}_i^{\mathcal{C}(h)}(\mathcal{P}_h(n)) &= \mathcal{P}_h(\text{suc}_i^h(n)) \quad \text{for all } n \in N, 0 \leq i < \text{ar}_h(n) \end{aligned}$$

The above definition follows a well-known approach to obtain, for each term graph g , a canonical representative $\mathcal{C}(g)$ [32]. One can easily see that $\mathcal{C}(g)$ is a well-defined canonical term graph. With this definition we indeed capture a notion of canonical representatives of isomorphism classes:

Proposition 4.16 (canonical term graphs are isomorphism class representatives). *Given $g \in \mathcal{G}^\infty(\Sigma)$, the term graph $\mathcal{C}(g)$ canonically represents the equivalence class $[g]_\cong$. More precisely, it holds that*

$$(i) [g]_\cong = [\mathcal{C}(g)]_\cong, \quad \text{and} \quad (ii) [g]_\cong = [h]_\cong \quad \text{iff} \quad \mathcal{C}(g) = \mathcal{C}(h).$$

In particular, we have, for all canonical term graphs g, h , that $g = h$ iff $g \cong h$.

Proof. Straightforward consequence of Lemma 4.13. \square

4.2.2. Labelled Quotient Trees. Intuitively, term graphs can be thought of as “terms with sharing”, i.e. terms in which occurrences of the same subterm may be identified. The representation of isomorphic term graphs as *labelled quotient trees*, which we shall study in this section, makes use of and formalises this intuition. To this end, we introduce an equivalence relation on the positions of a term graph that captures the sharing in a term graph:

Definition 4.17 (aliasing positions). Given a term graph g and two positions $\pi_1, \pi_2 \in \mathcal{P}(g)$, we say that π_1 and π_2 *alias each other* in g , denoted $\pi_1 \sim_g \pi_2$, if $\text{node}_g(\pi_1) = \text{node}_g(\pi_2)$.

One can easily see that the thus defined relation \sim_g on $\mathcal{P}(g)$ is an equivalence relation. Moreover, the partition on $\mathcal{P}(g)$ induced by \sim_g is simply the set $\{\mathcal{P}_g(n) \mid n \in N^g\}$ that contains the sets of positions of nodes in g .

Example 4.18. For the term graph g_2 illustrated in Figure 2a, we have that $\langle 0, 0 \rangle \sim_{g_2} \langle 1 \rangle$ as both $\langle 0, 0 \rangle$ and $\langle 1 \rangle$ are positions of the a -node in g_2 . For the term graph g_4 in Figure 2b, $\langle \rangle \sim_{g_4} \langle 1 \rangle \sim_{g_4} \langle 1, 1 \rangle \sim_{g_4} \dots$ as all finite sequences over 1 are positions of the f -node in g_4 .

The characterisation of Δ -homomorphisms of Lemma 4.10 can be recast in terms of aliasing positions, which then yields the following characterisation of the *existence* of Δ -homomorphisms:

Lemma 4.19 (characterisation of Δ -homomorphisms). *Given $g, h \in \mathcal{G}^\infty(\Sigma)$, there is a Δ -homomorphism $\phi: g \rightarrow_\Delta h$ iff, for all $\pi, \pi' \in \mathcal{P}(g)$, we have*

$$(a) \pi \sim_g \pi' \implies \pi \sim_h \pi', \quad \text{and} \quad (b) g(\pi) \notin \Delta \implies g(\pi) = h(\pi).$$

Proof. For the “only if” direction, assume that ϕ is a Δ -homomorphism from g to h . Then we can use the properties (a) and (b) of Lemma 4.10, which we will refer to as (a’) and (b’) to avoid confusion. In order to show (a), assume $\pi \sim_g \pi'$. Then there is some node $n \in N^g$ with $\pi, \pi' \in \mathcal{P}_g(n)$. (a’) yields $\pi, \pi' \in \phi(n)$ and, therefore, $\pi \sim_h \pi'$. To show (b), we assume some $\pi \in \mathcal{P}(g)$ with $g(\pi) \notin \Delta$. Then we can reason as follows:

$$g(\pi) = \text{lab}^g(\text{node}_g(\pi)) \stackrel{(b')}{=} \text{lab}^h(\phi(\text{node}_g(\pi))) \stackrel{(a')}{=} \text{lab}^h(\text{node}_h(\pi)) = h(\pi)$$

For the converse direction, assume that both (a) and (b) hold. Define the function $\phi: N^g \rightarrow N^h$ by $\phi(n) = m$ iff $\mathcal{P}_g(n) \subseteq \mathcal{P}_h(m)$ for all $n \in N^g$ and $m \in N^h$. To see that this is well-defined, we show at first that, for each $n \in N^g$, there is at most one $m \in N^h$ with $\mathcal{P}_g(n) \subseteq \mathcal{P}_h(m)$. Suppose there is another node $m' \in N^h$ with $\mathcal{P}_g(n) \subseteq \mathcal{P}_h(m')$. Since

$\mathcal{P}_g(n) \neq \emptyset$, this implies $\mathcal{P}_h(m) \cap \mathcal{P}_h(m') \neq \emptyset$. Hence, $m = m'$. Secondly, we show that there is at least one such node m . Choose some $\pi^* \in \mathcal{P}_g(n)$. Since then $\pi^* \sim_g \pi^*$ and, by (a), also $\pi^* \sim_h \pi^*$ holds, there is some $m \in N^h$ with $\pi^* \in \mathcal{P}_h(m)$. For each $\pi \in \mathcal{P}_g(n)$, we have $\pi^* \sim_g \pi$ and, therefore, $\pi^* \sim_h \pi$ by (a). Hence, $\pi \in \mathcal{P}_h(m)$. So we know that ϕ is well-defined. By construction, ϕ satisfies (a'). Moreover, because of (b), it is also easily seen to satisfy (b'). Hence, ϕ is a homomorphism from g to h . \square

Intuitively, Clause (a) states that h has at least as much sharing of nodes as g has, whereas Clause (b) states that h has at least the same non- Δ -labelling as g . In this sense, the above characterisation confirms the intuition about Δ -homomorphisms that we mentioned in Example 4.8, viz. Δ -homomorphisms may only introduce sharing and relabel Δ -nodes. This can be observed in the two Δ -homomorphisms illustrated in Figure 2.

From the above characterisations of the existence of Δ -homomorphisms, we can easily derive the following characterisation of Δ -isomorphisms using the uniqueness of Δ -homomorphisms between two term graphs:

Lemma 4.20 (characterisation of Δ -isomorphisms). *For all $g, h \in \mathcal{G}^\infty(\Sigma)$, $g \cong_\Delta h$ iff*

$$(a) \sim_g = \sim_h, \quad \text{and} \quad (b) g(\pi) = h(\pi) \text{ or } g(\pi), h(\pi) \in \Delta \quad \text{for all } \pi \in \mathcal{P}(g).$$

Proof. Immediate consequence of Lemma 4.19 and Proposition 4.9. \square

Remark 4.21. Δ -homomorphisms can be naturally lifted to the set of isomorphism classes $\mathcal{G}^\infty(\Sigma)/\cong$: we say that two Δ -homomorphisms $\phi: g \rightarrow_\Delta h$, $\phi': g' \rightarrow_\Delta h'$, are isomorphic, written $\phi \cong \phi'$ iff there are isomorphisms $\psi_1: g \cong g'$ and $\psi_2: h \cong h'$ such that $\psi_2 \circ \phi = \phi' \circ \psi_1$. Given a Δ -homomorphism $\phi: g \rightarrow_\Delta h$ in $\mathcal{G}^\infty(\Sigma)$, $[\phi]_\cong: [g]_\cong \rightarrow_\Delta [h]_\cong$ is a Δ -homomorphism in $\mathcal{G}^\infty(\Sigma)/\cong$. These Δ -homomorphisms then form a category which can easily be shown to be isomorphic to the category of Δ -homomorphisms on $\mathcal{G}_c^\infty(\Sigma)$ via the mapping $[\cdot]_\cong$.

Lemma 4.20 has shown that term graphs can be characterised up to isomorphism by only giving the equivalence \sim_g and the labelling $g(\cdot): \pi \mapsto g(\pi)$ of the involved term graphs. This observation gives rise to the following definition:

Definition 4.22 (labelled quotient trees). A *labelled quotient tree* over signature Σ is a triple (P, l, \sim) consisting of a non-empty set $P \subseteq \mathbb{N}^*$, a function $l: P \rightarrow \Sigma$, and an equivalence relation \sim on P that satisfies the following conditions for all $\pi, \pi' \in \mathbb{N}^*$ and $i \in \mathbb{N}$:

$$\pi \cdot \langle i \rangle \in P \implies \pi \in P \quad \text{and} \quad i < \text{ar}(l(\pi)) \quad (\text{reachability})$$

$$\pi \sim \pi' \implies \begin{cases} l(\pi) = l(\pi') & \text{and} \\ \pi \cdot \langle i \rangle \sim \pi' \cdot \langle i \rangle & \text{for all } i < \text{ar}(l(\pi)) \end{cases} \quad (\text{congruence})$$

In other words, a labelled quotient tree (P, l, \sim) is a ranked tree domain P together with a congruence \sim on it and a labelling function $l: P/\sim \rightarrow \Sigma$ that honours the rank. Also note that since P must be non-empty, the reachability condition implies that $\langle \rangle \in P$.

Example 4.23. The term graph g_2 depicted in Figure 2a is represented up to isomorphism by the labelled quotient tree (P, l, \sim) with $P = \{\langle \rangle, \langle 0 \rangle, \langle 0, 0 \rangle, \langle 1 \rangle\}$, $l(\langle \rangle) = f$, $l(\langle 0 \rangle) = h$, $l(\langle 0, 0 \rangle) = l(\langle 1 \rangle) = a$ and \sim the least equivalence relation on P with $\langle 0, 0 \rangle \sim \langle 1 \rangle$.

The following lemma confirms that labelled quotient trees uniquely characterise any term graph up to isomorphism:

Lemma 4.24 (labelled quotient trees are canonical). *Each term graph $g \in \mathcal{G}^\infty(\Sigma)$ induces a canonical labelled quotient tree $(\mathcal{P}(g), g(\cdot), \sim_g)$ over Σ . Vice versa, for each labelled quotient tree (P, l, \sim) over Σ there is a unique canonical term graph $g \in \mathcal{G}_C^\infty(\Sigma)$ whose canonical labelled quotient tree is (P, l, \sim) , i.e. $\mathcal{P}(g) = P$, $g(\pi) = l(\pi)$ for all $\pi \in P$, and $\sim_g = \sim$.*

Proof. The first part is trivial: $(\mathcal{P}(g), g(\cdot), \sim_g)$ satisfies the conditions from Definition 4.22.

For the second part, let (P, l, \sim) be a labelled quotient tree. Define the term graph $g = (N, \text{lab}, \text{suc}, r)$ by

$$\begin{aligned} N &= P / \sim & \text{lab}(n) &= f \quad \text{iff} \quad \exists \pi \in n. l(\pi) = f \\ r &= [\langle \rangle] \sim & \text{suc}_i(n) &= n' \quad \text{iff} \quad \exists \pi \in n. \pi \cdot \langle i \rangle \in n' \end{aligned}$$

The functions lab and suc are well-defined due to the congruence condition satisfied by (P, l, \sim) . Since P is non-empty and closed under prefixes, it contains $\langle \rangle$. Hence, r is well-defined. Moreover, by the reachability condition, each node in N is reachable from the root node. An easy induction proof shows that $\mathcal{P}_g(n) = n$ for each node $n \in N$. Thus, g is a well-defined canonical term graph. The canonical labelled quotient tree of g is obviously (P, l, \sim) . Whenever there are two canonical term graphs with the same canonical labelled quotient tree (P, l, \sim) , they are isomorphic due to Lemma 4.20 and, therefore, have to be identical by Proposition 4.16. \square

Labelled quotient trees provide a valuable tool for constructing canonical term graphs as we shall see. Nevertheless, the original graph representation remains convenient for practical purposes as it allows a straightforward formalisation of term graph rewriting and provides a finite representation of finite cyclic term graphs, which induce an infinite labelled quotient tree.

4.2.3. Terms, Term Trees & Unravelling. Before we continue, it is instructive to make the correspondence between terms and term graphs clear. First, note that, for each term tree t , the equivalence \sim_t is the identity relation $\mathcal{I}_{\mathcal{P}(t)}$ on $\mathcal{P}(t)$, i.e. $\pi_1 \sim_t \pi_2$ iff $\pi_1 = \pi_2$. Consequently, we have the following one-to-one correspondence between canonical term trees and terms: each term $t \in \mathcal{T}^\infty(\Sigma)$ induces the canonical term tree given by the labelled quotient tree $(\mathcal{P}(t), t(\cdot), \mathcal{I}_{\mathcal{P}(t)})$. For example, the term tree depicted in Figure 1a corresponds to the term $f(a, h(a, b))$. We thus consider the set of terms $\mathcal{T}^\infty(\Sigma)$ as the subset of canonical term trees of $\mathcal{G}_C^\infty(\Sigma)$.

With this correspondence in mind, we can define the *unravelling* of a term graph g as the unique term t such that there is a homomorphism $\phi: t \rightarrow g$. The unravelling of cyclic term graphs yields infinite terms, e.g. in Figure 8 on page 43, the term h_ω is the unravelling of the term graph g_2 . We use the notation $\mathcal{U}(g)$ for the unravelling of g .

5. A RIGID PARTIAL ORDER ON TERM GRAPHS

In this section, we shall establish a partial order suitable for formalising convergence of sequences of canonical term graphs similarly to p -convergence on terms.

Recall that p -convergence in term rewriting systems is based on a partial order \leq_\perp on the set $\mathcal{T}^\infty(\Sigma_\perp)$ of *partial* terms. The partial order \leq_\perp instantiates occurrences of \perp from left to right, i.e. $s \leq_\perp t$ iff t is obtained by replacing occurrences of \perp in s by arbitrary terms in $\mathcal{T}^\infty(\Sigma_\perp)$.

Since we are considering term graph rewriting as a generalisation of term rewriting, our aim is to generalise the partial order \leq_\perp on terms to term graphs. That is, the partial order we are looking for should coincide with \leq_\perp if restricted to term trees. Moreover, we also want to maintain the characteristic properties of the partial order \leq_\perp when generalising to term graphs. The most important characteristic we are striving for is a complete semilattice structure in order to define p -convergence in terms of the limit inferior. Apart from that, we also want to maintain the intuition of the partial order \leq_\perp , viz. the intuition of information preservation, which \leq_\perp captures on terms as we illustrated in Section 2. We will make this last guiding principle clearer as we go along.

Analogously to partial terms, we consider the class of *partial term graphs* simply as term graphs over the signature $\Sigma_\perp = \Sigma \uplus \{\perp\}$. In order to generalise the partial order \leq_\perp to term graphs, we need to formalise the instantiation of occurrences of \perp in term graphs. Δ -homomorphisms, for $\Delta = \{\perp\}$ – or \perp -homomorphisms for short – provide the right starting point for that. A homomorphism $\phi: g \rightarrow h$ maps each node in g to a node in h while preserving the *local* structure of each node, viz. its labelling and its successors. In the case of a \perp -homomorphism $\phi: g \rightarrow_\perp h$, the preservation of the labelling is suspended for nodes labelled \perp thus allowing ϕ to instantiate each \perp -node in g with an arbitrary node in h .

Therefore, we shall use \perp -homomorphisms as the basis for generalising \leq_\perp to canonical partial term graphs. This approach is based on the observation that \perp -homomorphisms characterise the partial order \leq_\perp on terms. Considering terms as canonical term trees, we obtain the following equivalence:

$$s \leq_\perp t \iff \text{there is a } \perp\text{-homomorphism } \phi: s \rightarrow_\perp t.$$

Thus, \perp -homomorphisms constitute the ideal tool for defining a partial order on canonical partial term graphs that generalises \leq_\perp . In the following subsection, we shall explore different partial orders on canonical partial term graphs based on \perp -homomorphisms.

5.1. Partial Orders on Term Graphs. Consider the *simple partial order* \leq_\perp^S defined on term graphs as follows: $g \leq_\perp^S h$ iff there is a \perp -homomorphism $\phi: g \rightarrow_\perp h$. This is a straightforward generalisation of the partial order \leq_\perp to term graphs. In fact, this partial order forms a complete semilattice on $\mathcal{G}_C^\infty(\Sigma_\perp)$ [10].

As we have explained in Section 2, p -convergence on terms is based on the ability of the partial order \leq_\perp to capture *information preservation* between terms – $s \leq_\perp t$ means that t contains at least the same information as s does. The limit inferior – and thus p -convergence – comprises the accumulated information that eventually remains stable. Following the approach on terms, a partial order suitable as a basis for convergence for term graph rewriting, has to capture an appropriate notion of information preservation as well.

One has to keep in mind, however, that term graphs encode an additional dimension of information through *sharing* of nodes, i.e. the fact that nodes may have multiple positions. Since \leq_\perp^S specialises to \leq_\perp on terms, it does preserve the information on the tree structure in the same way as \leq_\perp does. The difficult part is to determine the right approach to the role of sharing.

Indeed, \perp -homomorphisms instantiate occurrences of \perp and are thereby able to introduce new information. But while \perp -homomorphisms preserve the *local* structure of each node, they may change the *global* structure of a term graph by introducing sharing: for

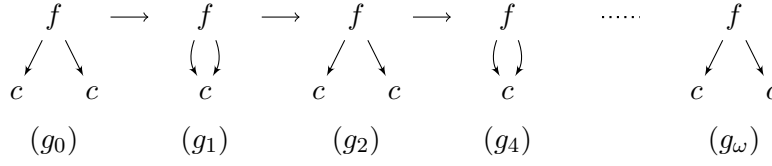
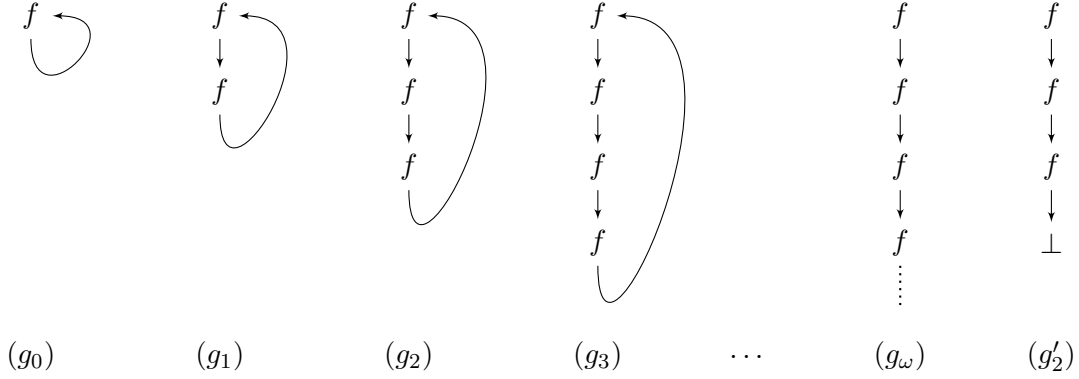

 FIGURE 3. Limit inferior w.r.t. \leq_{\perp}^S in the presence of acyclic sharing.


FIGURE 4. Varying acyclic sharing.

the term graphs g_0 and g_1 in Figure 3, we have an obvious \perp -homomorphism – in fact a homomorphism – $\phi: g_0 \rightarrow_{\perp} g_1$ and thus $g_0 \leq_{\perp}^S g_1$.

There are at least two different ways to interpret the differences in g_0 and g_1 . The first one dismisses \leq_{\perp}^S as a partial order suitable for our purposes: the term graphs g_0 and g_1 contain contradicting information. While in g_0 the two children of the f -node are distinct, they are identical in g_1 . We will indeed follow this view in this paper and introduce a rigid partial order \leq_{\perp}^R that addresses this concern. There is, however, also a second view that does not see g_0 and g_1 in contradiction: both term graphs show the f -node with two successors, both of which are labelled with c . The term graph g_1 merely contains the additional piece of information that the two successor nodes of the f -node are identical. The simple partial order \leq_{\perp}^S , which follows this view, is studied further in [10].

One consequence of the above behaviour of \leq_{\perp}^S is that total term graphs are not necessarily maximal w.r.t. \leq_{\perp}^S , e.g. g_0 is total but not maximal. The second – more severe – consequence is that there can be no metric on total term graphs such that the limit w.r.t. that metric coincides with the limit inferior on total term graph. To see this consider the sequence $(g_i)_{i < \omega}$ of term graphs illustrated in Figure 3. Its limit inferior w.r.t. \leq_{\perp}^S is the total term graph g_{ω} . On the other hand, there is no metric w.r.t. which $(g_i)_{i < \omega}$ converges since the sequence alternates between two distinct term graphs. That is, the correspondence between metric and partial order convergence that we know from term rewriting, cf. Theorem 3.3, is impossible.

In order to avoid the introduction of sharing, we need to consider \perp -homomorphisms that preserve the structure of term graphs more rigidly, i.e. not only locally. Recall that by Lemma 4.24, the structure of a term graph is essentially given by the positions of nodes

and their labelling. Labellings are already taken into consideration by \perp -homomorphisms. Thus, we can define a partial order \leq_{\perp}^P that preserves the structure of term graphs as follows: $g \leq_{\perp}^P h$ iff there is a \perp -homomorphism $\phi: g \rightarrow_{\perp} h$ with $\mathcal{P}_h(\phi(n)) = \mathcal{P}_g(n)$ for all $n \in N^g$ with $\text{lab}^g(n) \neq \perp$. While this would again yield a complete semilattice, it is unfortunately too restrictive. For example, consider the sequence of term graphs $(g_i)_{i < \omega}$ depicted in Figure 4. Due to the cycle, we have for each term graph g_i that \perp is the only term graph strictly smaller than g_i w.r.t. \leq_{\perp}^P . The reason for this is the fact that the only way to maintain the positions of the root node of the term graph g_i is to keep all nodes of the cycle in g_i . Hence, in order to obtain a term graph h with $h \leq_{\perp}^P g_i$, we have to either keep the whole term graph g_i or collapse it completely, yielding \perp . For example, we neither have $g'_2 \leq_{\perp}^P g_2$ nor $g'_2 \leq_{\perp}^P g_3$ for the term graph g'_2 illustrated in Figure 4. As a consequence, the limit inferior of the sequence $(g_i)_{i < \omega}$ is \perp and not the expected term graph g_{ω} .

The fact that the root nodes g_2 and g'_2 have different sets of positions is solely caused by the edge to the root node of g_2 that comes from *below* and thus closes a *cycle*. Even though the edge occurs below the root node, it affects its positions. Cutting off that edge, like in g'_2 , changes the sharing. As a consequence, in the complete semilattice $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^P)$, we do not obtain the intuitively expected convergence behaviour depicted in Figure 8c on page 43.

This observation suggests that we should only consider the *upward structure* of each node, ignoring the sharing that is caused by edges occurring *below* a node. We will see that by restricting our attention to *acyclic positions*, we indeed obtain the desired properties for a partial order on term graphs.

Recall that a position π in a term graph g is called cyclic iff there are positions π_1, π_2 with $\pi_1 < \pi_2 \leq \pi$ such that $\text{node}_g(\pi_1) = \text{node}_g(\pi_2)$, i.e. π passes a node twice. Otherwise it is called acyclic. We will use the notation $\mathcal{P}^a(g)$ for the set of all acyclic positions in g , and $\mathcal{P}_g^a(n)$ for the set of all acyclic positions of a node n in g . That is, $\mathcal{P}^a(g)$ is the set of positions in g that pass each node in g at most once. Clearly, every node has at least one acyclic position, i.e. $\mathcal{P}_g^a(n)$ is a non-empty set.

Definition 5.1 (rigidity). Let Σ be a signature, $\Delta \subseteq \Sigma^{(0)}$ and $g, h \in \mathcal{G}^{\infty}(\Sigma)$ such that $\phi: g \rightarrow_{\Delta} h$.

- (i) Given $n \in N^g$, ϕ is said to be *rigid* in n if it satisfies the equation

$$\mathcal{P}_g^a(n) = \mathcal{P}_h^a(\phi(n)) \quad (\text{rigid})$$

- (ii) ϕ is called a *rigid* Δ -homomorphism if it is rigid in all $n \in N^g$ with $\text{lab}^g(n) \notin \Delta$.

Proposition 5.2 (category of rigid Δ -homomorphisms). *The rigid Δ -homomorphisms on $\mathcal{G}^{\infty}(\Sigma)$ form a subcategory of the category of Δ -homomorphisms on $\mathcal{G}^{\infty}(\Sigma)$.*

Proof. Straightforward. □

Note that, for each node n in a term graph g , the positions in $\mathcal{P}_g^a(n)$ are minimal positions of n w.r.t. the prefix order. Rigid \perp -homomorphisms thus preserve the upward structure of each non- \perp -node and, therefore, provide the desired structure for a partial order that captures information preservation on term graphs:

Definition 5.3 (rigid partial order \leq_{\perp}^R). For every $g, h \in \mathcal{G}^{\infty}(\Sigma_{\perp})$, define $g \leq_{\perp}^R h$ iff there is a rigid \perp -homomorphism $\phi: g \rightarrow_{\perp} h$.

Proposition 5.4 (partial order \leq_{\perp}^R). *The relation \leq_{\perp}^R is a partial order on $\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp})$.*

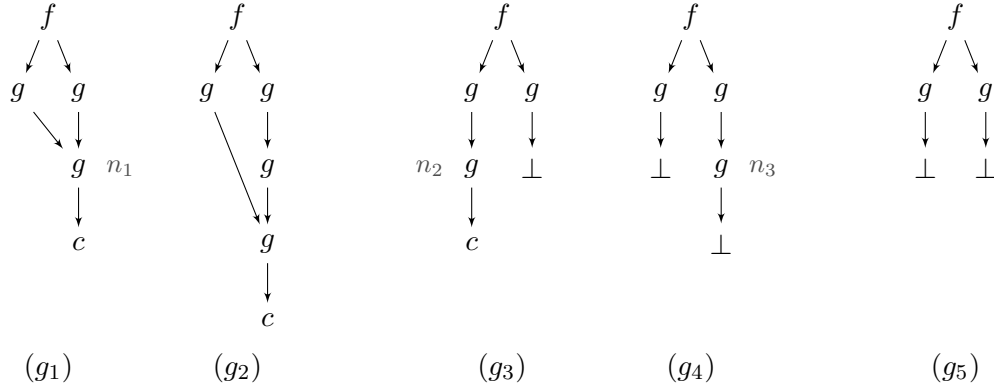


FIGURE 5. Term graphs g_1, g_2 with maximal lower bounds g_3, g_4 w.r.t. \leq_\perp^I .

Proof. Reflexivity and transitivity of \leq_\perp^R follow immediately from Proposition 5.2. For antisymmetry, assume $g \leq_\perp^R h$ and $h \leq_\perp^R g$. By Proposition 4.9, this implies $g \cong_\perp h$. Corollary 4.14 then yields that $g \cong h$. Hence, according to Proposition 4.16, $g = h$. \square

Example 5.5. Figure 8c on page 43 shows a sequence $(h_i)_{i < \omega}$ of term graphs and its limit inferior h_ω in $(\mathcal{G}_C^\infty(\Sigma_\perp), \leq_\perp^R)$: a cyclic list structure is repeatedly rewritten by inserting an element b in front of the a . We can see that in each step the newly inserted b (including the additional $::$ -node) remains unchanged afterwards. In terms of positions, however, each of the nodes changes in each step since the length of the cycle in the term graph grows with each step. Since this affects only cyclic positions, we still get the following sequence $(\bigcap_{\beta \leq i < \omega} h_i)_{\beta < \omega}$ of canonical term trees:

$$\langle \perp :: \perp, b :: \perp :: \perp, b :: b :: \perp :: \perp, \dots \rangle$$

The least upper bound of this sequence $(\bigcap_{\beta \leq i < \omega} h_i)_{\beta < \omega}$ and thus the limit inferior of $(h_i)_{i < \omega}$ is the infinite canonical term tree $h_\omega = b :: \bar{b} :: b :: \dots$. Since the cycle changes in each step and is thus cut through in each element of $(\bigcap_{\beta \leq i < \omega} h_i)_{\beta < \omega}$, the limit inferior has no cycles at all.

Note that we do not have this intuitively expected convergence behaviour for the partial order \leq_\perp^P based on positions: since the length of the cycle grows along the sequence $(h_i)_{i < \omega}$, we have that the set of positions of the root nodes changes constantly. Hence, the limit inferior of $(h_i)_{i < \omega}$ in $(\mathcal{G}_C^\infty(\Sigma_\perp), \leq_\perp^P)$ is \perp .

The partial order \leq_\perp^R based on rigid \perp -homomorphisms is defined in a rather non-local fashion as the definition of rigidity uses the set of *all* acyclic positions. This poses the question whether there is a more natural definition of a suitable partial order. One such candidate is the partial order \leq_\perp^I , which uses injectivity in order to restrict the introduction of sharing: $g \leq_\perp^I h$ iff there is a \perp -homomorphism $\phi: g \rightarrow_\perp h$ that is injective on non- \perp -nodes, i.e. $\phi(n) = \phi(m)$ and $\text{lab}^g(n), \text{lab}^g(m) \neq \perp$ implies $n = m$. While this yields indeed a cpo on $\mathcal{G}_C^\infty(\Sigma_\perp)$, we do not get a complete semilattice. To see this, consider Figure 5. The two term graphs g_3, g_4 are two distinct maximal lower bounds of the two term graphs g_1, g_2 w.r.t. the partial order \leq_\perp^I . Hence, the set $\{g_1, g_2\}$ does not have a greatest lower bound in $(\mathcal{G}_C^\infty(\Sigma_\perp), \leq_\perp^I)$, which is therefore not a complete semilattice. The same phenomenon occurs if we consider a partial order derived from \perp -homomorphisms that are injective on all nodes.

The rigid partial order \leq_{\perp}^R resolves the issue of \leq_{\perp}^I illustrated in Figure 5: g_3 and g_4 are not lower bounds of g_1 and g_2 w.r.t. \leq_{\perp}^R . The (unique) \perp -homomorphism from g_3 to g_1 is not rigid as it maps the node n_2 to n_1 and $\mathcal{P}_{g_3}^a(n_2) = \{\langle 0, 0 \rangle\}$ whereas $\mathcal{P}_{g_1}^a(n_1) = \{\langle 0, 0 \rangle, \langle 1, 0 \rangle\}$. Hence, $g_3 \not\leq_{\perp}^R g_1$. Likewise, $g_4 \not\leq_{\perp}^R g_1$ as the (unique) \perp -homomorphism from g_4 to g_1 maps n_3 to n_1 , which again have different acyclic positions. We do find, however, a greatest lower bound of g_1 and g_2 w.r.t. \leq_{\perp}^R , viz. g_5 .

5.2. The Rigid Partial Order. In the remainder of this section, we will study the rigid partial order \leq_{\perp}^R . In particular, we shall give a characterisation of rigidity in terms of labelled quotient trees analogous to Lemma 4.19, show that $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$ forms a complete semilattice, illustrate the resulting mode of convergence, and give a characterisation of term graphs that are maximal w.r.t. \leq_{\perp}^R .

The partial order \leq_{\perp}^I , derived from injective \perp -homomorphisms, failed to form a complete semilattice, which is why we abandoned that approach. The following lemma shows that rigidity is, in fact, a stronger property than injectivity on non- Δ -nodes. Hence, \leq_{\perp}^R is a restriction of \leq_{\perp}^I .

Lemma 5.6 (rigid Δ -homomorphisms are injective for non- Δ -nodes). *Let $g, h \in \mathcal{G}^{\infty}(\Sigma)$ and $\phi: g \rightarrow_{\Delta} h$ rigid. Then ϕ is injective for all non- Δ -nodes in g . That is, for two nodes $n, m \in N^g$ with $\text{lab}^g(n), \text{lab}^g(m) \notin \Delta$ we have that $\phi(n) = \phi(m)$ implies $n = m$.*

Proof. Let $n, m \in N^g$ with $\text{lab}^g(n), \text{lab}^g(m) \notin \Delta$ and $\phi(n) = \phi(m)$. Since ϕ is rigid, it is rigid in n and m . That is, in particular we have $\mathcal{P}_h^a(\phi(n)) \subseteq \mathcal{P}_g(n)$ and $\mathcal{P}_h^a(\phi(m)) \subseteq \mathcal{P}_g(m)$. Moreover, because $\mathcal{P}_h^a(\phi(n)) = \mathcal{P}_h^a(\phi(m)) \neq \emptyset$, we can conclude that $\mathcal{P}_g(n) \cap \mathcal{P}_g(m) \neq \emptyset$ and, therefore, $m = n$. \square

5.2.1. Characterising Rigidity. The goal of this subsection is to give a characterisation of rigidity in terms of labelled quotient trees. We will then combine this characterisation with Lemma 4.19 to obtain a characterisation of the partial order \leq_{\perp}^R .

The following lemma provides a characterisation of rigid Δ -homomorphisms that reduces the proof obligations necessary to show that a Δ -homomorphism is rigid.

Lemma 5.7 (rigidity). *Let $g, h \in \mathcal{G}^{\infty}(\Sigma)$, $\phi: g \rightarrow_{\Delta} h$. Then ϕ is rigid iff $\mathcal{P}_h^a(\phi(n)) \subseteq \mathcal{P}_g(n)$ for all $n \in N^g$ with $\text{lab}^g(n) \notin \Delta$.*

Proof. The “only if” direction is trivial. For the “if” direction, suppose that ϕ satisfies $\mathcal{P}_h^a(\phi(n)) \subseteq \mathcal{P}_g(n)$ for all $n \in N^g$ with $\text{lab}^g(n) \notin \Delta$. In order to prove that ϕ is rigid, we will show that $\mathcal{P}_h^a(\phi(n)) = \mathcal{P}_g^a(n)$ holds for each $n \in N^g$ with $\text{lab}^g(n) \notin \Delta$.

We first show the inclusion $\mathcal{P}_h^a(\phi(n)) \subseteq \mathcal{P}_g^a(n)$. For this purpose, let $\pi \in \mathcal{P}_h^a(\phi(n))$. Due to the hypothesis, this implies that $\pi \in \mathcal{P}_g(n)$. Now suppose that π is cyclic in g , i.e. there are two positions π_1, π_2 of a node $m \in N^g$ with $\pi_1 < \pi_2 \leq \pi$. By Lemma 4.10, we can conclude that $\pi_1, \pi_2 \in \mathcal{P}_h(\phi(m))$. This is a contradiction to the assumption that π is acyclic in h . Hence, $\pi \in \mathcal{P}_g^a(n)$.

For the other inclusion, assume some $\pi \in \mathcal{P}_g^a(n)$. Using Lemma 4.10 we obtain that $\pi \in \mathcal{P}_h(\phi(n))$. It remains to be shown that π is acyclic in h . Suppose that this is not true, i.e. there are two positions π_1, π_2 of a node $m \in N^h$ with $\pi_1 < \pi_2 \leq \pi$. Note that since $\pi \in \mathcal{P}(g)$, also $\pi_1, \pi_2 \in \mathcal{P}(g)$. Let $m_i = \text{node}_g(\pi_i)$, $i = 1, 2$. According to Lemma 4.10, we have that $\phi(m_1) = m = \phi(m_2)$. Moreover, observe that $g(\pi_1), g(\pi_2) \notin \Delta$: $g(\pi_1)$ cannot

be a nullary symbol because $\pi_1 < \pi \in \mathcal{P}(g)$. The same argument applies for the case that $\pi_2 < \pi$. If this is not the case, then $\pi_2 = \pi$ and $g(\pi) \notin \Delta$ follows from the assumption that $\text{lab}^g(n) \notin \Delta$. Thus, we can apply Lemma 5.6 to conclude that $m_1 = m_2$. Consequently, π is cyclic in g , which contradicts the assumption. Hence, $\pi \in \mathcal{P}_h^a(\phi(n))$. \square

From the above lemma we learn that Δ -isomorphisms are also rigid Δ -homomorphisms.

Corollary 5.8 (Δ -isomorphisms are rigid). *Let $g, h \in \mathcal{G}^\infty(\Sigma)$. If $\phi: g \cong_\Delta h$, then ϕ is a rigid Δ -homomorphism.*

Proof. This follows from Lemma 4.13 and Lemma 5.7. \square

Similarly to Lemma 4.19, we provide a characterisation of rigid Δ -homomorphisms in terms of labelled quotient trees:

Lemma 5.9 (characterisation of rigid Δ -homomorphisms). *Given $g, h \in \mathcal{G}^\infty(\Sigma)$, a Δ -homomorphism $\phi: g \rightarrow_\Delta h$ is rigid iff*

$$\pi \sim_h \pi' \implies \pi \sim_g \pi' \quad \text{for all } \pi \in \mathcal{P}(g) \text{ with } g(\pi) \notin \Delta \text{ and } \pi' \in \mathcal{P}^a(h).$$

Proof. For the “only if” direction, assume that ϕ is rigid. Moreover, let $\pi \in \mathcal{P}(g)$ with $g(\pi) \notin \Delta$ and $\pi' \in \mathcal{P}^a(h)$ such that $\pi \sim_h \pi'$, and let $n = \text{node}_g(\pi)$. By applying Lemma 4.10, we get that $\pi \in \mathcal{P}_h(\phi(n))$. Because of $\pi \sim_h \pi'$, also $\pi' \in \mathcal{P}_h(\phi(n))$. Since, according to the assumption, π' is acyclic in h , we know in particular that $\pi' \in \mathcal{P}_h^a(\phi(n))$. Since ϕ is rigid and $\text{lab}^g(n) \notin \Delta$, we know that ϕ is rigid in n which yields that $\pi' \in \mathcal{P}_g(n)$. Hence, $\pi \sim_g \pi'$.

For the converse direction, let $n \in N^g$ with $\text{lab}^g(n) \notin \Delta$. We need to show that ϕ is rigid in n . Due to Lemma 5.7, it suffices to show that $\mathcal{P}_h^a(\phi(n)) \subseteq \mathcal{P}_g(n)$. Since $\mathcal{P}_g(n) \neq \emptyset$, we can choose some $\pi^* \in \mathcal{P}_g(n)$. Then, according to Lemma 4.10, also $\pi^* \in \mathcal{P}_h(\phi(n))$. Let $\pi \in \mathcal{P}_h^a(\phi(n))$. Then $\pi^* \sim_h \pi$ holds. Since π is acyclic in h and $g(\pi^*) \notin \Delta$, we can use the hypothesis to obtain that $\pi^* \sim_g \pi$ holds which shows that $\pi \in \mathcal{P}_g(n)$. \square

Note that the above characterisation of rigidity is independent of the Δ -homomorphism at hand. This is expected since Δ -homomorphisms between a given pair of term graphs are unique.

By combining the above characterisation of rigidity with the corresponding characterisation of Δ -homomorphisms, we obtain the following compact characterisation of \leq_\perp^R :

Corollary 5.10 (characterisation of \leq_\perp^R). *Let $g, h \in \mathcal{G}^\infty(\Sigma_\perp)$. Then $g \leq_\perp^R h$ iff the following conditions are met:*

- (a) $\pi \sim_g \pi' \implies \pi \sim_h \pi' \quad \text{for all } \pi, \pi' \in \mathcal{P}(g)$
- (b) $\pi \sim_h \pi' \implies \pi \sim_g \pi' \quad \text{for all } \pi \in \mathcal{P}(g) \text{ with } g(\pi) \in \Sigma \text{ and } \pi' \in \mathcal{P}^a(h)$
- (c) $g(\pi) = h(\pi) \quad \text{for all } \pi \in \mathcal{P}(g) \text{ with } g(\pi) \in \Sigma$.

Proof. This follows immediately from Lemma 4.19 and Lemma 5.9. \square

Note that for term trees (b) is always true and (a) follows from (c). Hence, on term trees, \leq_\perp^R is characterised by (c) alone. This observation shows that \leq_\perp^R is indeed a generalisation of \leq_\perp .

Corollary 5.11. *For all $s, t \in \mathcal{T}^\infty(\Sigma_\perp)$, we have that $s \leq_\perp^R t$ iff $s \leq_\perp t$.*

Proof. Follows from Corollary 5.10. \square

5.2.2. *Convergence.* In the following, we shall show that \leq_\perp^R indeed forms a complete semi-lattice on $\mathcal{G}_C^\infty(\Sigma_\perp)$. We begin by showing that it constitutes a complete partial order.

Theorem 5.12 (\leq_\perp^R is a cpo). *The pair $(\mathcal{G}_C^\infty(\Sigma_\perp), \leq_\perp^R)$ forms a cpo. In particular, it has the least element \perp , and the least upper bound of a directed set G is given by the following labelled quotient tree (P, l, \sim) :*

$$P = \bigcup_{g \in G} \mathcal{P}(g) \quad \sim = \bigcup_{g \in G} \sim_g \quad l(\pi) = \begin{cases} f & \text{if } f \in \Sigma \text{ and } \exists g \in G. g(\pi) = f \\ \perp & \text{otherwise} \end{cases}$$

Proof. The least element of \leq_\perp^R is obviously \perp . Hence, it remains to be shown that each directed subset G of $\mathcal{G}_C^\infty(\Sigma_\perp)$ has a least upper bound w.r.t. \leq_\perp^R . To this end, we show that the canonical term graph \bar{g} given by the labelled quotient tree (P, l, \sim) described above is indeed the lub of G . We will make extensive use of Corollary 5.10 to do so. Therefore, we write (a), (b), (c) to refer to corresponding conditions of Corollary 5.10.

At first we need to show that l is indeed well-defined. For this purpose, let $g_1, g_2 \in G$ and $\pi \in \mathcal{P}(g_1) \cap \mathcal{P}(g_2)$ with $g_1(\pi), g_2(\pi) \in \Sigma$. Since G is directed, there is some $g \in G$ such that $g_1, g_2 \leq_\perp^R g$. By (c), we can conclude $g_1(\pi) = g(\pi) = g_2(\pi)$.

Next we show that (P, l, \sim) is indeed a labelled quotient tree. Recall that \sim needs to be an equivalence relation. For the reflexivity, assume that $\pi \in P$. Then there is some $g \in G$ with $\pi \in \mathcal{P}(g)$. Since \sim_g is an equivalence relation, $\pi \sim_g \pi$ must hold and, therefore, $\pi \sim \pi$. For the symmetry, assume that $\pi_1 \sim \pi_2$. Then there is some $g \in G$ such that $\pi_1 \sim_g \pi_2$. Hence, we get $\pi_2 \sim_g \pi_1$ and, consequently, $\pi_2 \sim \pi_1$. In order to show transitivity, assume that $\pi_1 \sim \pi_2, \pi_2 \sim \pi_3$. That is, there are $g_1, g_2 \in G$ with $\pi_1 \sim_{g_1} \pi_2$ and $\pi_2 \sim_{g_2} \pi_3$. Since G is directed, we find some $g \in G$ such that $g_1, g_2 \leq_\perp^R g$. By (a), this implies that also $\pi_1 \sim_g \pi_2$ and $\pi_2 \sim_g \pi_3$. Hence, $\pi_1 \sim_g \pi_3$ and, therefore, $\pi_1 \sim \pi_3$.

For the reachability condition, let $\pi \cdot \langle i \rangle \in P$. That is, there is a $g \in G$ with $\pi \cdot \langle i \rangle \in \mathcal{P}(g)$. Hence, $\pi \in \mathcal{P}(g)$, which in turn implies $\pi \in P$. Moreover, $\pi \cdot \langle i \rangle \in \mathcal{P}(g)$ implies that $i < \text{ar}(g(\pi))$. Since $g(\pi)$ cannot be a nullary symbol and in particular not \perp , we obtain that $l(\pi) = g(\pi)$. Hence, $i < \text{ar}(l(\pi))$.

For the congruence condition, assume that $\pi_1 \sim \pi_2$ and that $l(\pi_1) = f$. If $f \in \Sigma$, then there are $g_1, g_2 \in G$ with $\pi_1 \sim_{g_1} \pi_2$ and $g_2(\pi_1) = f$. Since G is directed, there is some $g \in G$ such that $g_1, g_2 \leq_\perp^R g$. Hence, by (a) respectively (c), we have $\pi_1 \sim_g \pi_2$ and $g(\pi_1) = f$. Using Lemma 4.24 we can conclude that $g(\pi_2) = g(\pi_1) = f$ and that $\pi_1 \cdot i \sim_g \pi_2 \cdot i$ for all $i < \text{ar}(g(\pi_1))$. Because $g \in G$, it holds that $l(\pi_2) = f$ and that $\pi_1 \cdot i \sim \pi \cdot i$ for all $i < \text{ar}(l(\pi_1))$. If $f = \perp$, then also $l(\pi_2) = \perp$, for if $l(\pi_2) = f'$ for some $f' \in \Sigma$, then, by the symmetry of \sim and the above argument (for the case $f \in \Sigma$), we would obtain $f = f'$ and, therefore, a contradiction. Since \perp is a nullary symbol, the remainder of the condition is vacuously satisfied.

This shows that (P, l, \sim) is a labelled quotient tree which, by Lemma 4.24, uniquely defines a canonical term graph. Next we show that the thus obtained term graph \bar{g} is an upper bound for G . To this end, let $g \in G$. We will show that $g \leq_\perp^R \bar{g}$ by establishing (a), (b) and (c). (a) and (c) are an immediate consequence of the construction. For (b), assume that $\pi_1 \in \mathcal{P}(g)$, $g(\pi_1) \in \Sigma$, $\pi_2 \in \mathcal{P}^a(\bar{g})$ and $\pi_1 \sim \pi_2$. We will show that then also $\pi_1 \sim_g \pi_2$ holds. Since $\pi_1 \sim \pi_2$, there is some $g' \in G$ with $\pi_1 \sim_{g'} \pi_2$. Because G is directed, there is some $g^* \in G$ with $g, g' \leq_\perp^R g^*$. Using (a), we then get that $\pi_1 \sim_{g^*} \pi_2$. Note that since π_2 is acyclic in \bar{g} , it is also acyclic in g^* : Suppose that this is not the case, i.e. there are positions π_3, π_4 with $\pi_3 < \pi_4 \leq \pi_2$ and $\pi_3 \sim_{g^*} \pi_4$. But then we also have $\pi_3 \sim \pi_4$,

which contradicts the assumption that π_2 is acyclic in \bar{g} . With this knowledge we are able to apply (b) to $\pi_1 \sim_{g^*} \pi_2$ in order to obtain $\pi_1 \sim_g \pi_2$.

In the final part of this proof, we will show that \bar{g} is the least upper bound of G . For this purpose, let \hat{g} be an upper bound of G , i.e. $g \leq_{\perp}^R \hat{g}$ for all $g \in G$. We will show that $\bar{g} \leq_{\perp}^R \hat{g}$ by establishing (a), (b) and (c). For (a), assume that $\pi_1 \sim \pi_2$. Hence, there is some $g \in G$ with $\pi_1 \sim_g \pi_2$. Since, by assumption, $g \leq_{\perp}^R \hat{g}$, we can conclude $\pi_1 \sim_{\hat{g}} \pi_2$ using (a). For (b), assume $\pi_1 \in P$, $l(\pi_1) \in \Sigma$, $\pi_2 \in \mathcal{P}^a(\hat{g})$ and $\pi_1 \sim_{\hat{g}} \pi_2$. That is, there is some $g \in G$ with $g(\pi_1) \in \Sigma$. Together with $g \leq_{\perp}^R \hat{g}$ this implies $\pi_1 \sim_g \pi_2$ by (b). $\pi_1 \sim \pi_2$ follows immediately. For (c), assume $\pi \in P$ and $l(\pi) = f \in \Sigma$. Then there is some $g \in G$ with $g(\pi) = f$. Applying (c) then yields $\hat{g}(\pi) = f$ since $g \leq_{\perp}^R \hat{g}$. \square

Remark 5.13. Following Remark 4.21, we define an order \leq_{\perp}^R on $\mathcal{G}^{\infty}(\Sigma_{\perp})/\cong$ which is isomorphic to the order \leq_{\perp}^R on $\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp})$. Define $[g]_{\cong} \leq_{\perp}^R [h]_{\cong}$ iff there is a rigid \perp -homomorphism $\phi: g \rightarrow_{\perp} h$.

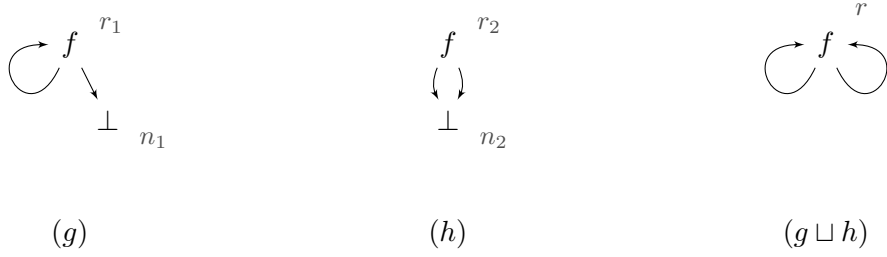
The extension of \leq_{\perp}^R to equivalence classes is easily seen to be well-defined: assume some rigid \perp -homomorphism $\phi: g \rightarrow_{\perp} h$ and two isomorphisms $g' \cong g$ and $h' \cong h$. Since, by Corollary 5.8, isomorphisms are also rigid (\perp -)homomorphisms, we have two rigid \perp -homomorphisms $\phi_1: g' \rightarrow_{\perp} g$ and $\phi_2: h \rightarrow_{\perp} h'$. Hence, by Proposition 5.2, $\phi_2 \circ \phi \circ \phi_1$ is a rigid \perp -homomorphism from g' to h' .

The isomorphism illustrated above allows us switch between the two partially ordered sets $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$ and $(\mathcal{G}^{\infty}(\Sigma_{\perp})/\cong, \leq_{\perp}^R)$ in order to use the structure that is more convenient for the given setting. In particular, the proof of Lemma 5.14 below is based on this isomorphism.

By Proposition 2.1, a cpo is a complete semilattice iff each two compatible elements have a least upper bound. Recall that compatible elements in a partially ordered set are elements that have a common upper bound. We make use of this proposition in order to show that $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$ is a complete semilattice. However, showing that each two term graphs $g, h \in \mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp})$ with a common upper bound also have a least upper bound is not easy. The issue that makes the construction of the lub of compatible term graphs a bit more complicated than in the case of directed sets is illustrated in Figure 6. Note that the lub $g \sqcup h$ of the term graphs g and h has an additional cycle. The fact that in $g \sqcup h$ the second successor of r has to be r itself is enforced by g saying that the first successor of r_1 is r_1 itself and by h saying that the first and the second successor of r_2 must be identical. Because of the additional cycle in $g \sqcup h$, we have that the set of positions in $g \sqcup h$ is a proper superset of the union of the sets of positions in g and h . This makes the construction of $g \sqcup h$ using a labelled quotient tree quite intricate.

Our strategy to construct the lub is to form the disjoint union of the two term graphs in question and then identify nodes that have a common position w.r.t. the term graph they originate from. In our example, we have four nodes r_1, n_1, r_2 and n_2 . At first r_1 and r_2 have to be identified as both have the position $\langle \rangle$. Next, r_1 and n_2 are identified as they share the position $\langle 0 \rangle$. And eventually, also n_2 and n_1 are identified since they share the position $\langle 1 \rangle$. Hence, all four nodes have to be identified. The result is, therefore, a term graph with a single node r . The following lemma and its proof, given in Appendix A, show that, for any two compatible term graphs, this construction always yields their lub.

Lemma 5.14 (compatible elements have lub). *Each pair g_1, g_2 of compatible term graphs in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$ has a least upper bound.*

FIGURE 6. Least upper bound $g \sqcup h$ of compatible term graphs g and h .

Theorem 5.15. *The pair $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$ forms a complete semilattice.*

Proof. This is, by Proposition 2.1, a consequence of Theorem 5.12 and Lemma 5.14. \square

In particular, this means that the limit inferior is defined for every sequence of term graphs.

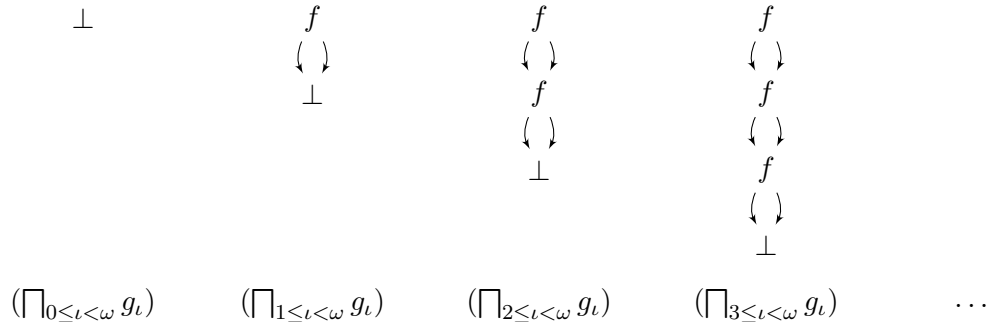
Corollary 5.16 (limit inferior of \leq_{\perp}^R). *Each sequence in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$ has a limit inferior.*

Recall that the intuition of the limit inferior on terms is that it contains the accumulated information that eventually remains stable in the sequence. This interpretation is, of course, based on the partial order \leq_{\perp} on terms, which embodies the underlying notion of “information encoded in a term”.

The same interpretation can be given for the limit inferior based on the rigid partial order \leq_{\perp}^R on term graphs. Given a sequence $(g_{\iota})_{\iota < \alpha}$ of term graphs, its limit inferior $\liminf_{\iota \rightarrow \alpha} g_{\iota}$ is the term graph that contains the accumulation of all pieces of information that from some point onwards remain unchanged in $(g_{\iota})_{\iota < \alpha}$.

Example 5.17. 9d and 9e on page 45 each show a sequence of term graphs and its limit inferior in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$.

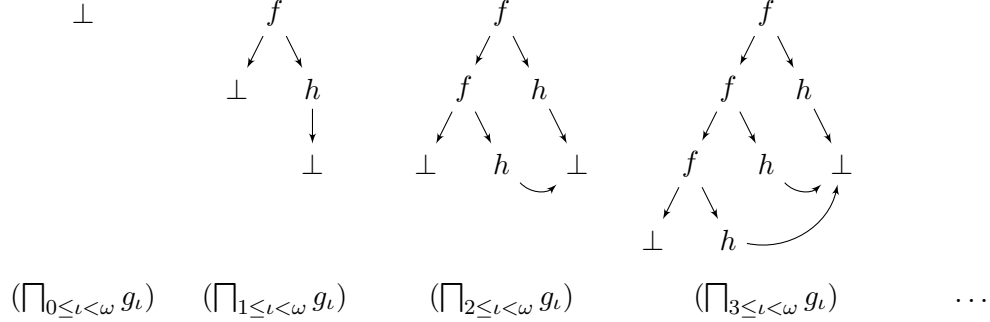
- (i) Figure 9d shows a simple example of how acyclic sharing is preserved by the limit inferior. The corresponding sequence $(\bigcap_{\beta \leq \iota < \omega} g_{\iota})_{\beta < \omega}$ of greatest lower bounds is given as follows:



The least upper bound of this sequence of term graphs and thus the limit inferior of $(g_{\iota})_{\iota < \omega}$ is the term graph g_{ω} depicted in Figure 9d.

- (ii) The situation is slightly different in the sequence $(g_{\iota})_{\iota < \omega}$ from Figure 9e. Here we also have acyclic sharing, viz. in the c -node. However, unlike in the previous example from Figure 9d, the acyclic sharing changes in each step. Hence, a lower bound

of two distinct term graphs in $(g_\iota)_{\iota < \omega}$ cannot contain a c -node because a rigid \perp -homomorphism must map such a c -node to a c -node with the same acyclic sharing, i.e. the same acyclic positions. Consequently, the sequence of greatest lower bounds $(\bigwedge_{\beta \leq \iota < \omega} g_\iota)_{\beta < \omega}$ looks as follows:



We thus get the term graph g_ω , depicted in Figure 9e, as the limit inferior of $(g_\iota)_{\iota < \omega}$. The \perp labelling is necessary because of the change in acyclic sharing throughout the sequence.

While we have confirmed in Corollary 5.11 that the partial order \leq_\perp^R generalises the partial order \leq_\perp on terms, we still have to show that this also carries over to the limit inferior. We can derive this property from the following simple lemma:

Lemma 5.18. *If $g \in \mathcal{G}_C^\infty(\Sigma_\perp)$ and $t \in \mathcal{T}^\infty(\Sigma_\perp)$ with $g \leq_\perp^R t$, then $g \in \mathcal{T}^\infty(\Sigma_\perp)$.*

Proof. Since t is a term tree, \sim_t is an identity relation. According to Corollary 5.10, $g \leq_\perp^R t$ implies that $\sim_g \subseteq \sim_t$. Hence, also \sim_g is an identity relation, which means that g is a term tree as well. \square

Proposition 5.19. *Given a sequence $(t_\iota)_{\iota < \alpha}$ over $\mathcal{T}^\infty(\Sigma_\perp)$, the limit inferior of $(t_\iota)_{\iota < \alpha}$ in $(\mathcal{T}^\infty(\Sigma_\perp), \leq_\perp)$ coincides with the limit inferior of $(t_\iota)_{\iota < \alpha}$ in $(\mathcal{G}_C^\infty(\Sigma_\perp), \leq_\perp^R)$.*

Proof. Since both structures are complete semilattices, both limit inferiors exist. For each $\beta < \alpha$, let s_β be the glb of $T_\beta = \{t_\iota \mid \beta \leq \iota < \alpha\}$ in $(\mathcal{T}^\infty(\Sigma_\perp), \leq_\perp)$ and g_β the glb of T_β in $(\mathcal{G}_C^\infty(\Sigma_\perp), \leq_\perp^R)$. Since then $g_\beta \leq_\perp^R t_\beta$, we know by Lemma 5.18 that g_β is a term tree. By Corollary 5.11, this implies that g_β is the glb of T_β in $(\mathcal{T}^\infty(\Sigma_\perp), \leq_\perp)$ as well, which means that $g_\beta = s_\beta$.

Let t and g be the limit inferior of $(t_\iota)_{\iota < \alpha}$ in $(\mathcal{T}^\infty(\Sigma_\perp), \leq_\perp)$ and $(\mathcal{G}_C^\infty(\Sigma_\perp), \leq_\perp^R)$, respectively. By the above argument, we know that t and g are the lub of the set $S = \{s_\beta \mid \beta < \alpha\}$ in $(\mathcal{T}^\infty(\Sigma_\perp), \leq_\perp)$ respectively $(\mathcal{G}_C^\infty(\Sigma_\perp), \leq_\perp^R)$. By Corollary 5.11, t is an upper bound of S in $(\mathcal{G}_C^\infty(\Sigma_\perp), \leq_\perp^R)$. Since g is the least such upper bound, we know that $g \leq_\perp^R t$. According to Lemma 5.18, this implies that g is a term tree. Hence, by Corollary 5.11, g is an upper bound of S in $(\mathcal{T}^\infty(\Sigma_\perp), \leq_\perp)$ and $g \leq_\perp t$. Since t is the least upper bound of S in $(\mathcal{T}^\infty(\Sigma_\perp), \leq_\perp)$, we can conclude that $t = g$. \square

5.2.3. Maximal Term Graphs. Intuitively, partial term graphs represent partial results of computations where \perp -nodes act as placeholders denoting the uncertainty or ignorance of the actual “value” at that position. On the other hand, total term graphs do contain all the information of a result of a computation – they have the maximally possible information

content. In other words, they are the maximal elements w.r.t. \leq_{\perp}^R . The following proposition confirms this intuition.

Proposition 5.20 (total term graphs are maximal). *Let Σ be a non-empty signature. Then $\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma)$ is the set of maximal elements in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$.*

Proof. At first we need to show that each element in $\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma)$ is maximal. For this purpose, let $g \in \mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma)$ and $h \in \mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp})$ such that $g \leq_{\perp}^R h$. We have to show that then $g = h$. Since $g \leq_{\perp}^R h$, there is a rigid \perp -homomorphism $\phi: g \rightarrow_{\perp} h$. As g does not contain any \perp -node, ϕ is even a rigid homomorphism. By Lemma 5.6, ϕ is injective and, therefore, according to Lemma 4.12, an isomorphism. Hence, we obtain that $g \cong h$ and, consequently, using Proposition 4.16, that $g = h$.

Secondly, we need to show that $\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp})$ does not contain any other maximal elements besides those in $\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma)$. Suppose there is a term graph $g \in \mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}) \setminus \mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma)$ which is maximal in $\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp})$. Hence, there is a node $n^* \in N^g$ with $\text{lab}^g(n^*) = \perp$. If Σ contains a nullary symbol c , construct a term graph h from g by relabelling the node n^* from \perp to c . However, then $g <_{\perp}^R h$, which contradicts the assumption that g is maximal w.r.t. \leq_{\perp}^R . Otherwise, if $\Sigma^{(0)} = \emptyset$, let \bar{n} be a fresh node (i.e. $\bar{n} \notin N^g$) and f some k -ary symbol in Σ . Define the term graph h by

$$\begin{aligned} N^h &= N^g \uplus \{\bar{n}\} & r^h &= r^g \\ \text{lab}^h(n) &= \begin{cases} f & \text{if } n = n^* \\ \perp & \text{if } n = \bar{n} \\ \text{lab}^g(n) & \text{otherwise} \end{cases} & \text{suc}^h(n) &= \begin{cases} \langle \bar{n}, \dots, \bar{n} \rangle & \text{if } n = n^* \\ \varepsilon & \text{if } n = \bar{n} \\ \text{suc}^g(n) & \text{otherwise} \end{cases} \end{aligned}$$

That is, h is obtained from g by relabelling n^* with f and setting the \perp -labelled node \bar{n} as the target of all outgoing edges of n^* . We assume that \bar{n} was chosen such that h is canonical (i.e. $\bar{n} = \mathcal{P}_h(\bar{n})$). Obviously, g and h are distinct. Define $\phi: N^g \rightarrow N^h$ by $n \mapsto n$ for all $n \in N^g$. Clearly, ϕ defines a rigid \perp -homomorphism from g to h . Hence, $g \leq_{\perp}^R h$. This contradicts the assumption of g being maximal. Consequently, no element in $\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}) \setminus \mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma)$ is maximal. \square

Note that this property does not hold for the simple partial order \leq_{\perp}^S that we have considered briefly in the beginning of this section. Figure 3 shows the total term graph g_0 , which is strictly smaller than g_1 w.r.t. \leq_{\perp}^S .

6. A RIGID METRIC ON TERM GRAPHS

In this section, we pursue the metric approach to convergence in rewriting systems. To this end, we shall define a metric space on canonical term graphs. We base our approach to defining a metric distance on the definition of the metric distance \mathbf{d} on terms. In particular, we shall define a *truncation* operation on term graphs, which cuts off certain nodes depending on their depth in the term graph. Subsequently, we study the interplay of the truncation with Δ -homomorphisms and the depth of nodes within a term graph. Finally, we use the truncation operation to derive a metric on term graphs.

6.1. Truncating Term Graphs. Originally, Arnold and Nivat [4] used a truncation of terms to define the metric on terms. The truncation of a term t at depth $d \leq \omega$, denoted $t|d$, replaces all subterms at depth d by \perp :

$$t|0 = \perp, \quad f(t_1, \dots, t_k)|d+1 = f(t_1|d, \dots, t_k|d), \quad t|\omega = t$$

Recall that the metric distance \mathbf{d} on terms is defined by $\mathbf{d}(s, t) = 2^{-\text{sim}(s, t)}$. The underlying notion of similarity $\text{sim}(\cdot, \cdot)$ can be characterised via truncations as follows:

$$\text{sim}(s, t) = \max \{d \leq \omega \mid s|d = t|d\}$$

We adopt this approach for term graphs as well. To this end, we shall define a rigid truncation on term graphs. In Section 6.3 we will then show that this truncation indeed yields a complete metric space.

Definition 6.1 (rigid truncation of term graphs). Let $g \in \mathcal{G}^\infty(\Sigma_\perp)$ and $d < \omega$.

- (i) Given $n, m \in N^g$, m is an *acyclic predecessor* of n in g if there is an acyclic position $\pi \cdot \langle i \rangle \in \mathcal{P}_g^a(n)$ with $\pi \in \mathcal{P}_g(m)$. The set of acyclic predecessors of n in g is denoted $\text{Pre}_g^a(n)$.
- (ii) The set of *retained nodes* of g at d , denoted $N_{<d}^g$, is the least subset M of N^g satisfying the following two conditions for all $n \in N^g$:

$$(T1) \text{ depth}_g(n) < d \implies n \in M \quad (T2) n \in M \implies \text{Pre}_g^a(n) \subseteq M$$

- (iii) For each $n \in N^g$ and $i \in \mathbb{N}$, we use n^i to denote a fresh node, i.e. $\{n^i \mid n \in N^g, i \in \mathbb{N}\}$ is a set of pairwise distinct nodes not occurring in N^g . The set of *fringe nodes* of g at d , denoted $N_{=d}^g$, is defined as the singleton set $\{r^g\}$ if $d = 0$, and otherwise as the set

$$\left\{ n^i \mid \begin{array}{l} n \in N_{<d}^g, 0 \leq i < \text{ar}_g(n) \text{ with } \text{succ}_i^g(n) \notin N_{<d}^g \\ \text{or } \text{depth}_g(n) \geq d-1, n \notin \text{Pre}_g^a(\text{succ}_i^g(n)) \end{array} \right\}$$

- (iv) The *rigid truncation* of g at d , denoted $g \ddagger d$, is the term graph defined by

$$\begin{aligned} N^{g \ddagger d} &= N_{<d}^g \uplus N_{=d}^g & r^{g \ddagger d} &= r^g \\ \text{lab}^{g \ddagger d}(n) &= \begin{cases} \text{lab}^g(n) & \text{if } n \in N_{<d}^g \\ \perp & \text{if } n \in N_{=d}^g \end{cases} & \text{succ}_i^{g \ddagger d}(n) &= \begin{cases} \text{succ}_i^g(n) & \text{if } n^i \notin N_{=d}^g \\ n^i & \text{if } n^i \in N_{=d}^g \end{cases} \end{aligned}$$

Additionally, we define $g \ddagger \omega$ to be the term graph g itself.

Before discussing the intuition behind this definition of rigid truncation, let us have a look at the rôle of retained and fringe nodes: the set of retained nodes $N_{<d}^g$ contains the nodes that are *preserved* by the rigid truncation. All other nodes in $N^g \setminus N_{<d}^g$ are cut off. The “holes” that are thus created are filled by the fringe nodes in $N_{=d}^g$. This is expressed in the condition $\text{succ}_i^g(n) \notin N_{<d}^g$ which, if satisfied, yields a fringe node n^i . That is, a fresh fringe node is inserted for each successor of a retained node that is not a retained node itself. As fringe nodes function as a replacement for cut-off sub-term graphs, they are labelled with \perp and have no successors.

But there is another circumstance that can give rise to a fringe node: if $\text{depth}_g(n) \geq d-1$ and $n \notin \text{Pre}_g^a(\text{succ}_i^g(n))$, we also get a fringe node n^i . This condition is satisfied whenever an outgoing edge from a retained node closes a cycle. The lower bound for the depth is chosen such that a successor node of n is not necessarily a retained node. An example is depicted in Figure 7a. For depth $d = 2$, the node n in the term graph g is just above the fringe, i.e. satisfies $\text{depth}_g(n) \geq d-1$. Moreover, it has an edge to the node r that closes a cycle.

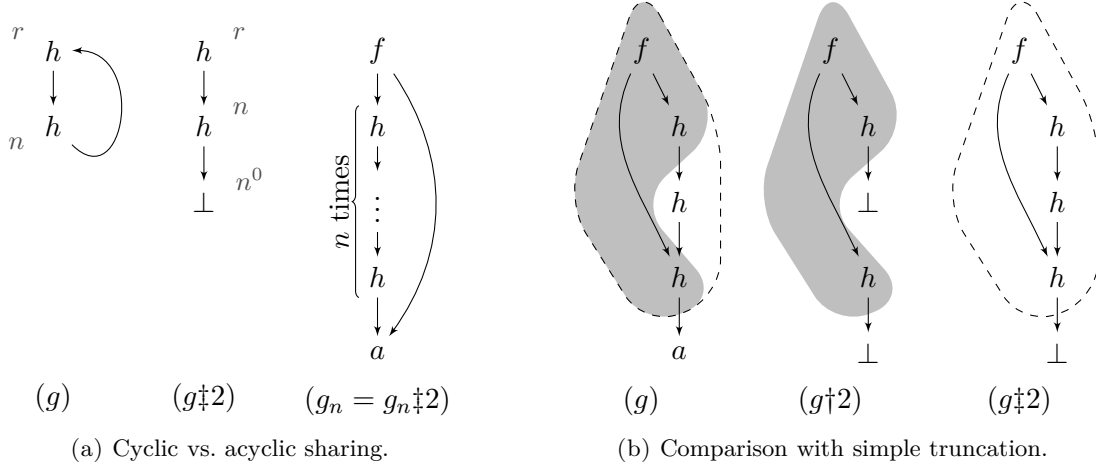


FIGURE 7. Examples of truncations.

Hence, the rigid truncation $g \ddagger 2$ contains the fringe node n^0 which is now the 0-th successor of n .

We chose this admittedly complicated notion of truncation in order to make it compatible with the partial order \leq_{\perp}^R : first of all, the rigid truncation of a term graph is supposed to yield a smaller term graph w.r.t. the rigid partial order \leq_{\perp}^R , i.e. $g \ddagger d \leq_{\perp}^R g$. Hence, whenever a node is kept as a retained node, also its acyclic positions have to be kept in order to preserve its upward structure. To achieve this, with each node also its *acyclic ancestors* have to be retained. The closure condition (T2) is enforced exactly for this purpose.

To see what this means, consider Figure 7b. It shows a term graph g and its truncation at depth 2, once without the closure condition (T2), denoted $g \dagger 2$, and once including (T2), denoted $g \ddagger 2$. The grey area highlights the nodes that are at depth smaller than 2, i.e. the nodes contained in $N_{<2}^g$ due to (T1). The nodes within the area surrounded by a dashed line are all the nodes in $N_{<2}^g$. One can observe that with the *simple truncation* $g \dagger d$ without (T2), we do not have $g \dagger 2 \leq_{\perp}^R g$. The reason in this particular example is the bottommost h -node whose acyclic sharing in g differs from that in the simple truncation $g \dagger 2$ as one of its predecessors was removed due to the truncation. This effect is avoided in our definition of rigid truncation, which always includes all acyclic predecessors of a node.

Nevertheless, the simple truncation $g \dagger d$ has its benefits. It is much easier to work with and provides a natural counterpart for the simple partial order \leq_{\perp}^S [10].

The following lemma confirms that we were indeed successful in making the truncation of term graphs compatible with the rigid partial order \leq_{\perp}^R :

Lemma 6.2 (rigid truncation is smaller). *Given $g \in \mathcal{G}^\infty(\Sigma_{\perp})$ and $d \leq \omega$, we have that $g \ddagger d \leq_{\perp}^R g$.*

Proof. The cases $d = \omega$ and $d = 0$ are trivial. Assume $0 < d < \omega$ and define the function ϕ as follows:

$$\begin{aligned} \phi: N^{g \ddagger d} &\rightarrow N^g \\ N_{<d}^g \ni n &\mapsto n \\ N_{=d}^g \ni n^i &\mapsto \text{succ}_i^g(n) \end{aligned}$$

We will show that ϕ is a rigid \perp -homomorphism from $g\dagger d$ to g and, thereby, $g\dagger d \leq_{\perp}^R g$.

Since $r^{g\dagger d} = r^g$ and $r^{g\dagger d} \in N_{<d}^g$, we have $\phi(r^{g\dagger d}) = r^g$ and, therefore, the root condition. Note that all nodes in $N_{=d}^g$ are labelled with \perp in $g\dagger d$, i.e. all non- \perp -nodes are in $N_{<d}^g$. Thus, the labelling condition is trivially satisfied as for all $n \in N_{<d}^g$ we have

$$\text{lab}^{g\dagger d}(n) = \text{lab}^g(n) = \text{lab}^g(\phi(n)).$$

For the successor condition, let $n \in N_{<d}^g$. If $n^i \in N_{=d}^g$, then $\text{suc}_i^{g\dagger d}(n) = n^i$. Hence, we have

$$\phi(\text{suc}_i^{g\dagger d}(n)) = \phi(n^i) = \text{suc}_i^g(n) = \text{suc}_i^g(\phi(n)).$$

If, on the other hand, $n^i \notin N_{=d}^g$, then $\text{suc}_i^{g\dagger d}(n) = \text{suc}_i^g(n) \in N_{<d}^g$. Hence, we have

$$\phi(\text{suc}_i^{g\dagger d}(n)) = \phi(\text{suc}_i^g(n)) = \text{suc}_i^g(n) = \text{suc}_i^g(\phi(n)).$$

This shows that ϕ is a \perp -homomorphism. In order to prove that ϕ is rigid, we will show that $\mathcal{P}_g^a(\phi(n)) \subseteq \mathcal{P}_{g\dagger d}(n)$ for all $n \in N_{<d}^g$, which is sufficient according to Lemma 5.7. Note that we can replace $\phi(n)$ by n since $n \in N_{<d}^g$. Therefore, we can show this statement by proving

$$\forall \pi \in \mathbb{N}^* \forall n \in N_{<d}^g. (\pi \in \mathcal{P}_g^a(n) \implies \pi \in \mathcal{P}_{g\dagger d}(n))$$

by induction on the length of π . If $\pi = \langle \rangle$, then $n = r^g$ and, therefore, $\pi \in \mathcal{P}_{g\dagger d}(n)$. If $\pi = \pi' \cdot \langle i \rangle$, let $m = \text{node}_g(\pi')$. Then we have $m \in \text{Pre}_g^a(n)$ and, therefore, $m \in N_{<d}^g$ by the closure property (T2). And since $\pi' \in \mathcal{P}_g^a(m)$, we can apply the induction hypothesis to obtain that $\pi' \in \mathcal{P}_{g\dagger d}(m)$. Moreover, because $\text{suc}_i^g(m) = n$, this implies that $m^i \notin N_{=d}^g$. Thus, $\text{suc}_i^{g\dagger d}(m) = n$ and, therefore, $\pi' \cdot \langle i \rangle \in \mathcal{P}_{g\dagger d}(n)$. \square

Also note that the rigid truncation on term graphs generalises Arnold and Nivat's [4] truncation on terms.

Proposition 6.3. *For each $t \in \mathcal{T}^\infty(\Sigma_\perp)$ and $d \leq \omega$, we have that $t\dagger d \cong t|d$.*

Proof. For the case that $d \in \{0, \omega\}$, the equation $t\dagger d = t|d$ holds trivially. For the other cases, we can easily see that $t|d$ is obtained from t by replacing all subterms at depth d by \perp . On the other hand, since in a term tree each node has at most one (acyclic) predecessor, which has a strictly smaller depth, we know that the set of retained nodes $N_{<d}^t$ is the set of nodes of depth smaller than d and the set of fringe nodes $N_{=d}^t$ is the set $\{n^i \mid n \in N^t, \text{depth}_t(\text{suc}_i(n)) = d\}$. Hence, $t\dagger d$ is obtained from t by replacing each node at depth d with a fresh node labelled \perp . We can thus conclude that $t\dagger d \cong t|d$. \square

Consequently, if we use the rigid truncation to define a metric on term graphs analogously to Arnold and Nivat, we obtain a metric on term graphs that generalises the metric **d** on terms.

6.2. The Effect of Truncation. In order to characterise the effect of a truncation to a term graph, we need to associate an appropriate notion of depth to a whole term graph:

Definition 6.4 (symbol/graph depth). Let $g \in \mathcal{G}^\infty(\Sigma)$ and $\Delta \subseteq \Sigma$.

- (i) The *depth* of g , denoted $\text{depth}(g)$, is the least upper bound of the depth of nodes in g , i.e.

$$\text{depth}(g) = \bigsqcup \left\{ \text{depth}_g(n) \mid n \in N^g \right\}.$$

- (ii) The Δ -depth of g , denoted $\Delta\text{-depth}(g)$, is the minimum depth of nodes in g labelled in Δ , i.e.

$$\Delta\text{-depth}(g) = \min \left\{ \text{depth}_g(n) \mid n \in N^g, \text{lab}^g(n) \in \Delta \right\} \cup \{\omega\}.$$

If Δ is a singleton set $\{\sigma\}$, we also write $\sigma\text{-depth}(g)$ instead of $\{\sigma\}\text{-depth}(g)$.

Notice the difference between depth and Δ -depth. The former is the least upper bound of the depth of nodes in a term graph whereas the latter is the *minimum* depth of nodes labelled by a symbol in Δ . Thus, we have that $\text{depth}(g) = \omega$ iff g is infinite; and $\Delta\text{-depth}(g) = \omega$ iff g does not contain a Δ -node.

In the following, we will prove a number of lemmas that show how Δ -homomorphisms preserve the depth of nodes in term graphs. Understanding how Δ -homomorphisms affect the depth of nodes will become important for relating the rigid truncation to the rigid partial order \leq_{\perp}^R .

Lemma 6.5 (reverse depth preservation of Δ -homomorphisms). *Let $g, h \in \mathcal{G}^\infty(\Sigma)$ and $\phi: g \rightarrow_{\Delta} h$. Then, for all $n \in N^h$ with $\text{depth}_h(n) \leq \Delta\text{-depth}(g)$, there is a node $m \in \phi^{-1}(n)$ with $\text{depth}_g(m) \leq \text{depth}_h(n)$.*

Proof. We prove the statement by induction on $\text{depth}_h(n)$. If $\text{depth}_h(n) = 0$, then $n = r^h$. With $m = r^g$, we have $\phi(m) = n$ and $\text{depth}_g(m) = 0$. If $\text{depth}_h(n) > 0$, then there is some $n' \in N^h$ with $\text{suc}_i^h(n') = n$ and $\text{depth}_h(n') < \text{depth}_h(n)$. Hence, we can employ the induction hypothesis to obtain some $m' \in N^g$ with $\text{depth}_g(m') \leq \text{depth}_h(n')$ and $\phi(m') = n'$. Since $\text{depth}_g(m') \leq \text{depth}_h(n') < \text{depth}_h(n) \leq \Delta\text{-depth}(g)$, we have $\text{lab}^g(m') \notin \Delta$. Hence, ϕ is homomorphic in m' . For $m = \text{suc}_i^g(m')$, we can then reason as follows:

$$\begin{aligned} \phi(m) &= \phi(\text{suc}_i^g(m')) = \text{suc}_i^h(\phi(m')) = \text{suc}_i^h(n') = n, \quad \text{and} \\ \text{depth}_g(m) &\leq \text{depth}_g(m') + 1 \leq \text{depth}_h(n). \end{aligned} \quad \square$$

Lemma 6.6 (Δ -depth preservation of Δ -homomorphisms). *Let $g, h \in \mathcal{G}^\infty(\Sigma)$ and $\phi: g \rightarrow_{\Delta} h$, then $\Delta\text{-depth}(g) \leq \Delta\text{-depth}(h)$.*

Proof. Let $n \in N^h$ with $\text{depth}_h(n) < \Delta\text{-depth}(g)$. To prove the lemma, we have to show that $\text{lab}^h(n) \notin \Delta$. According to Lemma 6.5, we find a node $m \in N^g$ with $\text{depth}_g(m) \leq \text{depth}_h(n) < \Delta\text{-depth}(g)$ and $\phi(m) = n$. Since then $\text{lab}^g(m) \notin \Delta$, we also have $\text{lab}^h(n) \notin \Delta$ by the labelling condition for ϕ . \square

For rigid Δ -homomorphisms, we even have a stronger form of depth preservation.

Lemma 6.7 (depth preservation of rigid Δ -homomorphisms). *Let $g, h \in \mathcal{G}^\infty(\Sigma)$ and $\phi: g \rightarrow_{\Delta} h$ a rigid Δ -homomorphism. Then $\text{depth}_g(n) = \text{depth}_h(\phi(n))$ for all $n \in N^g$ with $\text{lab}^g(n) \notin \Delta$.*

Proof. If $\text{lab}^g(n) \notin \Delta$, then $\mathcal{P}_g^a(n) = \mathcal{P}_h^a(\phi(n))$. Hence, $\text{depth}_g(n) = \text{depth}_h(\phi(n))$ follows since a shortest position of a node must be acyclic. \square

The gaps that are caused by a truncation due to the removal of nodes are filled by fresh \perp -nodes. The following lemma provides a lower bound for the depth of the introduced \perp -nodes.

Lemma 6.8 (\perp -depth in rigid truncations). *For all $g \in \mathcal{G}^\infty(\Sigma)$ and $d < \omega$, we have that*

- (i) $\perp\text{-depth}(g \upharpoonright d) \geq d$, and

(ii) if $d > \text{depth}(g) + 1$, then $g \dagger d = g$, i.e. $\perp\text{-depth}(g \dagger d) = \omega$.

Proof. (i) From the proof of Lemma 6.2, we obtain a rigid \perp -homomorphism $\phi: g \dagger d \rightarrow_{\perp} g$. Note that the only \perp -nodes in $g \dagger d$ are those in $N_{=d}^g$. Each of these nodes has only a single predecessor, a node $n \in N_{<d}^g$ with $\text{depth}_g(n) \geq d - 1$. By Lemma 6.7, we also have $\text{depth}_{g \dagger d}(n) \geq d - 1$ for these nodes since ϕ is rigid, n is not labelled with \perp and $\phi(n) = n$. Hence, we have $\text{depth}_{g \dagger d}(m) \geq d$ for each node $m \in N_{=d}^g$. Consequently, it holds that $\perp\text{-depth}(g \dagger d) \geq d$.

(ii) Note that if $d > \text{depth}(g) + 1$, then $N_{<d}^g = N^g$ and $N_{=d}^g = \emptyset$. Hence, $g \dagger d = g$. \square

Remark 6.9. Note that the precondition for the statement of clause (ii) in the lemma above reads $d > \text{depth}(g) + 1$ rather than $d > \text{depth}(g)$ as one might expect. The reason for this is that a truncation might cut off an edge that emanates from a node at depth $d - 1$ and closes a cycle. For an example of this phenomenon, take a look at Figure 7a. It shows a term graph g of depth 1 and its rigid truncation at depth 2. Even though there is no node at depth 2 the truncation introduces a \perp -node.

On the other hand, although a term graph has depth greater than d , the truncation at depth d might still preserve the whole term graph. An example for this behaviour is the family of term graphs $(g_n)_{n < \omega}$ depicted in Figure 7a. Each of the term graphs g_n has depth $n + 1$. Yet, the truncation at depth 2 preserves the whole term graph g_n for each $n > 0$. Even though there might be h -nodes which are at depth ≥ 2 these nodes are directly or indirectly acyclic predecessors of the a -node and are, thus, included in $N_{<2}^{g_n}$.

Intuitively, the following lemma states that a rigid \perp -homomorphism has the properties of an isomorphism up to the depth of the shallowest \perp -node:

Lemma 6.10 (\leq_{\perp}^R and rigid truncation). *Given $g, h \in \mathcal{G}^{\infty}(\Sigma_{\perp})$ and $d < \omega$ with $g \leq_{\perp}^R h$ and $\perp\text{-depth}(g) \geq d$, we have that $g \dagger d \cong h \dagger d$.*

The proof of the above lemma is based on a generalisation of Lemma 6.7, which states that rigid \perp -homomorphisms map non- \perp -nodes to nodes of the same depth. However, since the rigid truncation of a term graph does not only depend on the depth of nodes but also the acyclic sharing in the term graph, we cannot rely on this statement on the depth of nodes alone. The two key components of the proof of Lemma 6.10 are (1) the property of rigid \perp -homomorphisms to map retained nodes of the source term graph exactly to the retained nodes of the target term graph and (2) that in the same way fringe nodes are exactly mapped to fringe nodes. Showing the isomorphism between $g \dagger d$ and $h \dagger d$ can thus be reduced to the injectivity on retained nodes in $g \dagger d$ which is obtained from the rigid \perp -homomorphism from g to h by applying Lemma 5.6. The full proof of Lemma 6.10 is given in Appendix B.

We can use the above findings in order to obtain the following properties of truncations that one would intuitively expect from a truncation operation:

Lemma 6.11 (smaller truncations). *For all $g, h \in \mathcal{G}^{\infty}(\Sigma)$ and $e \leq d \leq \omega$, the following holds:*

$$(i) \ g \dagger e \cong (g \dagger d) \dagger e, \quad \text{and} \quad (ii) \ g \dagger d \cong h \dagger d \implies g \dagger e \cong h \dagger e.$$

Proof. (i) For $d = \omega$, this is trivial. Suppose $d < \omega$. From Lemma 6.2, we obtain $g \dagger d \leq_{\perp}^R g$. Moreover, by Lemma 6.8, we have $\perp\text{-depth}(g \dagger d) \geq d$ and, a fortiori, $\perp\text{-depth}(g \dagger d) \geq e$. Hence, we can employ Lemma 6.10 to get $(g \dagger d) \dagger e \cong g \dagger e$.

(ii) Since $g \dot{\vdash} d \cong h \dot{\vdash} d$, we also have $(g \dot{\vdash} d) \dot{\vdash} e \cong (h \dot{\vdash} d) \dot{\vdash} e$, as the construction of the truncation only depends on the structure of the term graphs. Hence, using (i) we can conclude

$$g \dot{\vdash} e \cong (g \dot{\vdash} d) \dot{\vdash} e \cong (h \dot{\vdash} d) \dot{\vdash} e \cong h \dot{\vdash} e. \quad \square$$

6.3. Deriving a Metric on Term Graphs. We may now define a rigid distance measure on canonical term graphs in the style of Arnold and Nivat:

Definition 6.12 (rigid distance). The *rigid similarity* of two term graphs $g, h \in \mathcal{G}_C^\infty(\Sigma)$, written $\text{sim}_\dagger(g, h)$, is the maximum depth at which the rigid truncation of both term graphs coincide:

$$\text{sim}_\dagger(g, h) = \max \{d \leq \omega \mid g \dot{\vdash} d \cong h \dot{\vdash} d\}.$$

The *rigid distance* between two term graphs $g, h \in \mathcal{G}_C^\infty(\Sigma)$, written $\mathbf{d}_\dagger(g, h)$ is defined as

$$\mathbf{d}_\dagger(g, h) = 2^{-\text{sim}_\dagger(g, h)}, \text{ where we interpret } 2^{-\omega} \text{ as } 0.$$

Indeed, the resulting distance forms an ultrametric on the set of canonical term graphs:

Proposition 6.13 (rigid ultrametric). *The pair $(\mathcal{G}_C^\infty(\Sigma), \mathbf{d}_\dagger)$ forms an ultrametric space.*

Proof. The identity condition is derived as follows:

$$\mathbf{d}_\dagger(g, h) = 0 \iff \text{sim}_\dagger(g, h) = \omega \iff g \cong h \xrightarrow{\text{Prop. 4.16}} g = h$$

The symmetry condition is satisfied by the following equational reasoning:

$$\mathbf{d}_\dagger(g, h) = 2^{-\text{sim}_\dagger(g, h)} = 2^{-\text{sim}_\dagger(h, g)} = \mathbf{d}_\dagger(h, g)$$

For the strong triangle condition, we have to show that

$$\mathbf{d}_\dagger(g_1, g_3) \leq \max \{\mathbf{d}_\dagger(g_1, g_2), \mathbf{d}_\dagger(g_2, g_3)\},$$

which is equivalent to

$$\text{sim}_\dagger(g_1, g_3) \geq \min \{\text{sim}_\dagger(g_1, g_2), \text{sim}_\dagger(g_2, g_3)\}.$$

Let $d = \text{sim}_\dagger(g_1, g_2)$ and $e = \text{sim}_\dagger(g_2, g_3)$. By symmetry, we can assume w.l.o.g. that $d \leq e$, i.e. $d = \min \{\text{sim}_\dagger(g_1, g_2), \text{sim}_\dagger(g_2, g_3)\}$. By definition of rigid similarity, we have both $g_1 \dot{\vdash} d \cong g_2 \dot{\vdash} d$ and $g_2 \dot{\vdash} e \cong g_3 \dot{\vdash} e$. From the latter we obtain, by Lemma 6.11, that $g_2 \dot{\vdash} d \cong g_3 \dot{\vdash} d$. That is, $g_1 \dot{\vdash} d \cong g_2 \dot{\vdash} d \cong g_3 \dot{\vdash} d$ which means that $\text{sim}_\dagger(g_1, g_3) \geq d$. \square

Example 6.14. Figures 8c and 9d on pages 43 and 45, respectively, show two sequences of term graphs that are converging in the metric space $(\mathcal{G}_C^\infty(\Sigma), \mathbf{d}_\dagger)$. In the sequence $(h_i)_{i < \omega}$ from Figure 8c, we have that the rigid truncation at 0 is trivially \perp for all term graphs in the sequence. From h_1 onwards, the rigid truncation at 1 is the term tree $\perp :: \perp$; from h_2 onwards, the rigid truncation at 2 is the term tree $b :: \perp :: \perp$; etc. Hence, for each $n < \omega$, the metric distance $\mathbf{d}_\dagger(h_i, h_j)$ between two term graphs from h_n onwards, i.e. with $n \leq i, j < \omega$, is at most 2^{-n} . That is, the sequence $(h_i)_{i < \omega}$ is Cauchy. Even more, for the term tree $h_\omega = b :: b :: b :: \dots$ depicted in Figure 8c we also have that $h_\omega \dot{\vdash} 0 = \perp$, $h_\omega \dot{\vdash} 1 = \perp :: \perp$, $h_\omega \dot{\vdash} 2 = b :: \perp :: \perp$, etc. Hence, for each $n < \omega$, the metric distance $\mathbf{d}_\dagger(h_n, h_\omega)$ is at most 2^{-n} . That is, the sequence $(h_i)_{i < \omega}$ converges to h_ω . In a similar fashion, the sequence depicted in Figure 9d converges as well.

Figure 9e shows a sequence $(g_i)_{i < \omega}$ of term graphs that does not converge. In fact, it is not even Cauchy. To see this, notice that the c -node is at depth 1 in g_0 and at depth 2 from g_1 onwards. As in each term graph g_i the c -node is reachable from any node in g_i without

forming a cycle, we have that each node is an acyclic ancestor of the c -node. That is, whenever the c -node is retained by a rigid truncation, so is any other node. Consequently, we have that $g_i \dagger d = g_i$ for each $i < \omega$ and $d > 2$. Hence, the metric distance $\mathbf{d}_{\dagger}(g_i, g_j)$ between each pair of term graphs with $i \neq j$ is at least 2^{-2} . That is, $(g_i)_{i < \omega}$ is not Cauchy.

Since we defined the metric on term graphs in the same style as Arnold and Nivat [4] defined the partial order \mathbf{d} on terms, we can use the correspondence between the rigid truncation and the truncation on terms in order to derive that the metric \mathbf{d}_{\dagger} generalises the metric \mathbf{d} on terms.

Corollary 6.15. *For all $s, t \in \mathcal{T}^{\infty}(\Sigma)$, we have that $\mathbf{d}_{\dagger}(s, t) = \mathbf{d}(s, t)$.*

Proof. Follows from Proposition 6.3. \square

From the above observation, we obtain that convergence in the metric space $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma), \mathbf{d}_{\dagger})$ is a conservative extension of convergence in the metric space $(\mathcal{T}^{\infty}(\Sigma), \mathbf{d})$:

Proposition 6.16. *Every non-empty sequence over $\mathcal{T}^{\infty}(\Sigma)$ converges to t in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma), \mathbf{d}_{\dagger})$ iff it converges to t in $(\mathcal{T}^{\infty}(\Sigma), \mathbf{d})$.*

Proof. The “if” direction follows immediately from Corollary 6.15.

For the “iff” direction we assume a sequence S over $\mathcal{T}^{\infty}(\Sigma)$ that converges to t in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma), \mathbf{d}_{\dagger})$. Consequently, S is also Cauchy in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma), \mathbf{d}_{\dagger})$. Due to Corollary 6.15, S is then also Cauchy in $(\mathcal{T}^{\infty}(\Sigma), \mathbf{d})$. Since $(\mathcal{T}^{\infty}(\Sigma), \mathbf{d})$ is complete, S converges to some term t' in $(\mathcal{T}^{\infty}(\Sigma), \mathbf{d})$. Using the “if” direction of this proposition, we then obtain that S converges to t' in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma), \mathbf{d}_{\dagger})$. Since limits are unique in metric spaces, we can conclude that $t = t'$. \square

7. METRIC VS. PARTIAL ORDER CONVERGENCE

In this section we study both the partially ordered set $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$ and the metric space $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma), \mathbf{d}_{\dagger})$. In particular, we are interested in the notion of convergence that each of the two structures provides. We shall show that on total term graphs – i.e. in $\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma)$ – both structures yield the same notion of convergence. That is, we obtain the same correspondence that we already know from infinitary term rewriting as stated in Theorem 3.3. Moreover, as a side product, this finding will also show the completeness of the metric space $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma), \mathbf{d}_{\dagger})$.

The cornerstone of this comparison of the rigid metric \mathbf{d}_{\dagger} and the rigid partial order \leq_{\perp}^R is the following characterisation of the rigid similarity $\text{sim}_{\dagger}(\cdot, \cdot)$ in terms of greatest lower bounds in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$:

Proposition 7.1 (characterisation of rigid similarity). *Let $g, h \in \mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma)$ and $g \sqcap h$ the greatest lower bound of g and h in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$. Then $\text{sim}_{\dagger}(g, h) = \perp\text{-depth}(g \sqcap h)$.*

Proof. At first assume that $g = h$. Hence, $g \sqcap h = g$ and, consequently $\perp\text{-depth}(g \sqcap h) = \omega$ as g does not contain any node labelled \perp . On the other hand, $g = h$ implies $g \dagger \omega \cong h \dagger \omega$, and, therefore, $\text{sim}_{\dagger}(g, h) = \omega$. If $g \neq h$, then $g \not\sqsubseteq h$ by Proposition 4.16. Hence, $\text{sim}_{\dagger}(g, h) < \omega$. Moreover, since $g \not\sqsubseteq h$, we know that $g \sqcap h$ has to be strictly smaller than g or h w.r.t. \leq_{\perp}^R . Hence, according to Proposition 5.20, $g \sqcap h$ has to contain some node labelled \perp , i.e. $\perp\text{-depth}(g \sqcap h) < \omega$ as well. We prove that $\perp\text{-depth}(g \sqcap h) = \text{sim}_{\dagger}(g, h)$ by showing that both $\perp\text{-depth}(g \sqcap h) \leq \text{sim}_{\dagger}(g, h)$ and $\perp\text{-depth}(g \sqcap h) \geq \text{sim}_{\dagger}(g, h)$ hold.

In order to show the former, let $d = \perp\text{-depth}(g \sqcap h)$. Since $g \sqcap h \leq_{\perp}^R g, h$, we can apply Lemma 6.10 twice in order to obtain $g \dagger d \cong (g \sqcap h) \dagger d \cong h \dagger d$. Hence, $\text{sim}_{\dagger}(g, h) \geq d$.

To show the converse direction, let $d = \text{sim}_{\dagger}(g, h)$, i.e. $g \dagger d \cong h \dagger d$. According to Lemma 6.2, we have both $g \dagger d \leq_{\perp}^R g$ and $h \dagger d \leq_{\perp}^R h$. Note that, for the canonical representation, we then have $\mathcal{C}(g \dagger d) = \mathcal{C}(h \dagger d)$, $\mathcal{C}(g \dagger d) \leq_{\perp}^R g$ and $\mathcal{C}(h \dagger d) \leq_{\perp}^R h$ (cf. Proposition 4.16 respectively Remark 5.13). That is, $\mathcal{C}(g \dagger d)$ is a lower bound of g and h . Thus, $\mathcal{C}(g \dagger d) \leq_{\perp}^R g \sqcap h$ and we can reason as follows:

$$d \leq \perp\text{-depth}(g \dagger d) \quad (\text{Lem. 6.8})$$

$$= \perp\text{-depth}(\mathcal{C}(g \dagger d)) \quad (\text{Lem. 6.7, Cor. 5.8})$$

$$\leq \perp\text{-depth}(g \sqcap h) \quad (\mathcal{C}(g \dagger d) \leq_{\perp}^R g \sqcap h, \text{Lem. 6.6})$$

□

Remark 7.2. From now on, we are not dealing with the concrete construction of rigid truncations $g \dagger d$ anymore. Therefore, we will rather use the canonical representation $\mathcal{C}(g \dagger d)$ of $g \dagger d$. In order to avoid the notational overhead, we write $g \dagger d$ instead of $\mathcal{C}(g \dagger d)$.

In the next step we show that the metric space $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma), \mathbf{d}_{\dagger})$ is indeed complete. The following proposition states even more: the limit of Cauchy sequences in the metric space equals its limit inferior in the partially ordered set $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$:

Proposition 7.3 (metric limit = limit inferior). *Let $(g_{\iota})_{\iota < \alpha}$ be a non-empty Cauchy sequence in the metric space $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma), \mathbf{d}_{\dagger})$ and $\liminf_{\iota \rightarrow \alpha} g_{\iota}$ its limit inferior in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$. Then $\lim_{\iota \rightarrow \alpha} g_{\iota} = \liminf_{\iota \rightarrow \alpha} g_{\iota}$.*

Proof. If α is a successor ordinal, this is trivial, as the limit and the limit inferior are then $g_{\alpha-1}$. Assume that α is a limit ordinal and let \bar{g} be the limit inferior of $(g_{\iota})_{\iota < \alpha}$. Since, according to Theorem 5.15, $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$ is a complete semilattice, \bar{g} is well-defined. Since $(g_{\iota})_{\iota < \alpha}$ is Cauchy, we obtain that, for each $\varepsilon \in \mathbb{R}^+$, there is a $\beta < \alpha$ such that we have $\mathbf{d}_{\dagger}(g_{\iota}, g_{\iota'}) < \varepsilon$ for all $\beta \leq \iota, \iota' < \alpha$. A fortiori, we get that, for each $\varepsilon \in \mathbb{R}^+$, there is a $\beta < \alpha$ such that we have $\mathbf{d}_{\dagger}(g_{\beta}, g_{\iota}) < \varepsilon$ for all $\beta \leq \iota < \alpha$. By definition of \mathbf{d}_{\dagger} , this is equivalent to $2^{-\text{sim}_{\dagger}(g_{\beta}, g_{\iota})} < \varepsilon$. Consequently, we have, for each $d < \omega$, a $\beta < \alpha$ such that $\text{sim}_{\dagger}(g_{\beta}, g_{\iota}) > d$ for all $\beta \leq \iota < \alpha$. Due to Lemma 6.11, $\text{sim}_{\dagger}(g_{\beta}, g_{\iota}) > d$ implies $g_{\beta} \dagger d = g_{\iota} \dagger d$ which in turn implies $g_{\beta} \dagger d \leq_{\perp}^R g_{\iota}$, according to Lemma 6.2. Hence, $g_{\beta} \dagger d$ is a lower bound for $G_{\beta} = \{g_{\iota} \mid \beta \leq \iota < \alpha\}$, i.e. $g_{\beta} \dagger d \leq_{\perp}^R \bigcap G_{\beta}$. Moreover, by the definition of the limit inferior, it holds that $\bigcap G_{\beta} \leq_{\perp}^R \bar{g}$. Consequently, $g_{\beta} \dagger d \leq_{\perp}^R \bar{g}$, i.e. we have

$$\forall d < \omega \exists \beta < \alpha: \quad g_{\beta} \dagger d \leq_{\perp}^R \bar{g} \quad (1)$$

Since, by Lemma 6.8, we have $\perp\text{-depth}(g_{\beta} \dagger d) \geq d$, we can apply Lemma 6.10 to obtain $(g_{\beta} \dagger d) \dagger d \cong \bar{g} \dagger d$. Hence, by Lemma 6.11, we have $g_{\beta} \dagger d \cong \bar{g} \dagger d$ and therefore $\text{sim}_{\dagger}(\bar{g}, g_{\beta}) \geq d$. That is, we have shown that

$$\forall d < \omega \exists \beta < \alpha: \quad \text{sim}_{\dagger}(\bar{g}, g_{\beta}) \geq d$$

Since, for each $\varepsilon \in \mathbb{R}^+$, we find a $d < \omega$ with $2^{-d} < \varepsilon$, this implies

$$\forall \varepsilon \in \mathbb{R}^+ \exists \beta < \alpha: \quad \mathbf{d}_{\dagger}(\bar{g}, g_{\beta}) < \varepsilon$$

This shows that $(g_{\iota})_{\iota < \alpha}$ converges to \bar{g} . Now it remains to be shown that \bar{g} is indeed in $\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma)$, i.e. it does not contain any node labelled \perp . Suppose that \bar{g} does contain a node labelled with \perp . Then $\perp\text{-depth}(\bar{g}) < \omega$. Let $d = \perp\text{-depth}(\bar{g}) + 1$. By (1), there is a β with $g_{\beta} \dagger d \leq_{\perp}^R \bar{g}$. By applying Lemma 6.8 and Lemma 6.6, we then get

$$\perp\text{-depth}(\bar{g}) + 1 = d \leq \perp\text{-depth}(g_{\beta} \dagger d) \leq \perp\text{-depth}(\bar{g}).$$

This is a contradiction. Hence, \bar{g} is indeed in $\mathcal{G}_C^\infty(\Sigma)$. \square

This result has two obvious but important consequences: firstly, the limit of a converging sequence in the rigid metric space is equal to the limit inferior in the rigid complete semilattice. Secondly, the rigid metric space $(\mathcal{G}_C^\infty(\Sigma), \mathbf{d}_\dagger)$ is complete:

Theorem 7.4 (completeness of rigid metric). *The metric space $(\mathcal{G}_C^\infty(\Sigma), \mathbf{d}_\dagger)$ is complete.*

Proof. Immediate consequence of Proposition 7.3. \square

In the following proposition, we show the converse direction of the relation between the limits of the rigid metric and the limit inferiors of the rigid partial order:

Proposition 7.5 (total limit inferior = limit). *Let $(g_\iota)_{\iota < \alpha}$ be a non-empty sequence in $(\mathcal{G}_C^\infty(\Sigma), \mathbf{d}_\dagger)$ and $\liminf_{\iota \rightarrow \alpha} g_\iota$ its limit inferior in $(\mathcal{G}_C^\infty(\Sigma_\perp), \leq_\perp^R)$. If $\liminf_{\iota \rightarrow \alpha} g_\iota \in \mathcal{G}_C^\infty(\Sigma)$, then $\liminf_{\iota \rightarrow \alpha} g_\iota = \lim_{\iota \rightarrow \alpha} g_\iota$.*

Proof. If α is a successor ordinal, then both the limit and the limit inferior are equal to $g_{\alpha-1}$. Let α be a limit ordinal. According to Proposition 7.3, in order to show that $(g_\iota)_{\iota < \alpha}$ converges and that its limit coincides with its limit inferior, it suffices to prove that $(g_\iota)_{\iota < \alpha}$ is Cauchy.

Let $\bar{g} = \liminf_{\iota \rightarrow \alpha} g_\iota$, and define $G_\beta = \{g_\iota \mid \beta \leq \iota < \alpha\}$ and $h_\beta = \bigcap G_\beta$ for each $\beta < \alpha$. Note that $\bar{g} = \bigcup_{\beta < \alpha} h_\beta$. Since \bar{g} is total, i.e. no node in \bar{g} is labelled with \perp , we have, according to Theorem 5.12, that for each $\pi \in \mathcal{P}(\bar{g})$ there is some $\beta_\pi < \alpha$ with $h_{\beta_\pi}(\pi) = \bar{g}(\pi)$.

Note that $(h_\iota)_{\iota < \alpha}$ is monotonic w.r.t. \leq_\perp^R , i.e. $\iota \leq \iota'$ implies $h_\iota \leq_\perp^R h_{\iota'}$. Since $h_\iota \leq_\perp^R h_{\iota'}$ together with $h_\iota(\pi) \in \Sigma$ implies $h_{\iota'}(\pi) = h_\iota(\pi)$ by Corollary 5.10, we have $h_\gamma(\pi) = \bar{g}(\pi)$ for all $\pi \in \mathcal{P}(\bar{g})$ and $\beta_\pi \leq \gamma < \alpha$.

Let $d < \omega$. Since there are only finitely many positions in $\mathcal{P}(\bar{g})$ of length smaller than d , we know that $\beta_d = \max \{\beta_\pi \mid \pi \in \mathcal{P}(\bar{g}), |\pi| < d\}$ is a well-defined ordinal smaller than α . Hence, for all $\pi \in \mathcal{P}(\bar{g})$ with $|\pi| < d$ we have $h_{\beta_d}(\pi) = \bar{g}(\pi)$. Since \bar{g} is total, we thus have that $\perp\text{-depth}(h_{\beta_d}) \geq d$.

Since $g_\iota, g_{\iota'} \in G_{\beta_d}$ for each $\beta_d \leq \iota, \iota' < \alpha$, we have $h_{\beta_d} \leq_\perp^R g_\iota, g_{\iota'}$ and thus $h_{\beta_d} \leq_\perp^R g_\iota \sqcap g_{\iota'}$. Consequently, by Lemma 6.6, we have that $\perp\text{-depth}(g_\iota \sqcap g_{\iota'}) \geq \perp\text{-depth}(h_{\beta_d})$. That is,

$$\text{sim}_\dagger(g_\iota, g_{\iota'}) \stackrel{\text{Prop. 7.1}}{=} \perp\text{-depth}(g_\iota \sqcap g_{\iota'}) \geq \perp\text{-depth}(h_{\beta_d}) \geq d \quad \text{for each } \beta_d < \iota, \iota' < \alpha.$$

Now, let $\varepsilon \in \mathbb{R}^+$. We then find some $d < \omega$ with $\varepsilon > 2^{-d}$. Consequently, we have

$$\mathbf{d}_\dagger(g_\iota, g_{\iota'}) = 2^{-\text{sim}_\dagger(g_\iota, g_{\iota'})} \leq 2^{-d} < \varepsilon \quad \text{for all } \beta_d \leq \iota, \iota' < \alpha.$$

Hence, $(g_\iota)_{\iota < \alpha}$ is Cauchy. \square

Note that Proposition 7.5 depends on the finiteness of the arity of the symbols in the signature. (This is used in the proof above when observing that a term graph has only finitely many positions of a bounded length.) This restriction also applies to terms as the following example shows:

Example 7.6. Let $\Sigma = \{f/\omega, a/0, b/0\}$ and $(t_i)_{i < \omega}$ a sequence with

$$\begin{aligned} t_0 &= f(a, a, a, a, a \dots), \\ t_1 &= f(b, a, a, a, a \dots), \\ t_2 &= f(b, b, a, a, a \dots), \\ t_3 &= f(b, b, b, a, a \dots), \\ &\vdots \end{aligned}$$

$(t_i)_{i < \omega}$ has the limit inferior $f(b, b, b, b, b, \dots)$. On the other hand, the sequence is not even Cauchy since, for each $i \neq j$, we have $\text{sim}_{\ddagger}(t_i, t_j) = 1$ and, therefore, $\mathbf{d}_{\ddagger}(t_i, t_j) = \frac{1}{2}$.

8. INFINITARY TERM GRAPH REWRITING

In the previous sections, we have constructed and investigated the necessary metric and partial order structures upon which the infinitary calculus of term graph rewriting that we shall introduce in this section is based. After describing the framework of term graph rewriting that we consider, we will explore two different modes of convergence on term graphs. In the same way that infinitary term rewriting is based on the abstract notions of m - and p -convergence [6], infinitary term graph rewriting is an instantiation of these abstract modes of convergence for term graphs. However, as in the overview of infinitary term rewriting in Section 2, we restrict ourselves to weak notions of convergence.

8.1. Term Graph Rewriting Systems. In this paper, we adopt the term graph rewriting framework of Barendregt et al. [11]. In order to represent placeholders in rewrite rules, this framework uses variables – in a manner much similar to term rewrite rules. To this end, we consider a signature $\Sigma_{\mathcal{V}} = \Sigma \uplus \mathcal{V}$ that extends the signature Σ with a set \mathcal{V} of nullary variable symbols.

Definition 8.1 (term graph rewriting systems).

- (i) Given a signature Σ , a *term graph rule* ρ over Σ is a triple (g, l, r) where g is a graph over $\Sigma_{\mathcal{V}}$ and $l, r \in N^g$ such that all nodes in g are reachable from l or r . We write ρ_l respectively ρ_r to denote the left- respectively right-hand side of ρ , i.e. the term graph $g|_l$ respectively $g|_r$. Additionally, we require that, for each variable $v \in \mathcal{V}$, there is at most one node n in g labelled v and that n is different but still reachable from l .
- (ii) A *term graph rewriting system (GRS)* \mathcal{R} is a pair (Σ, R) with Σ a signature and R a set of term graph rules over Σ .

The requirement that the root l of the left-hand side is not labelled with a variable symbol is analogous to the requirement that the left-hand side of a term rule is not a variable. Similarly, the restriction that nodes labelled with variable symbols must be reachable from the root of the left-hand side corresponds to the restriction on term rules that every variable occurring on the right-hand side of a rule must also occur on the left-hand side.

Term graphs can be used to compactly represent terms, which is formalised by the unravelling operator $\mathcal{U}(\cdot)$. We extend this operator to term graph rules. Figure 8a illustrates two term graph rules that both represent the term rule $a :: x \rightarrow b :: a :: x$ from Example 3.1 to which they unravel.

Definition 8.2 (unravelling of term graph rules). Let ρ be a term graph rule with ρ_l and ρ_r its left- respectively right-hand side term graph. The *unravelling* of ρ , denoted $\mathcal{U}(\rho)$ is the term rule $\mathcal{U}(\rho_l) \rightarrow \mathcal{U}(\rho_r)$.

The application of a rewrite rule ρ (with root nodes l and r) to a term graph g is performed in three steps: at first a suitable sub-term graph of g rooted in some node n of g is *matched* against the left-hand side of ρ . This amounts to finding a \mathcal{V} -homomorphism $\phi: \rho_l \rightarrow_{\mathcal{V}} g|_n$ from the term graph rooted in l to the sub-term graph rooted in n , the *redex*. The \mathcal{V} -homomorphism ϕ allows us to instantiate variables in the rule with sub-term graphs of the redex. In the second step, nodes and edges in ρ that are not reachable from l are copied into g , such that each edge pointing to a node m in the term graph rooted in l is redirected to $\phi(m)$. In the last step, all edges pointing to n are redirected to (the copy of) r and all nodes not reachable from the root of (the now modified version of) g are removed.

The formal definition of this construction is given below:

Definition 8.3 (application of term graph rewrite rules, [11]). Let $\rho = (N^\rho, \text{lab}^\rho, \text{suc}^\rho, l^\rho, r^\rho)$ be a term graph rewrite rule in a GRS $\mathcal{R} = (\Sigma, R)$, $g \in \mathcal{G}^\infty(\Sigma)$ with $N^\rho \cap N^g = \emptyset$ and $n \in N^g$. ρ is called *applicable* to g at n if there is a \mathcal{V} -homomorphism $\phi: \rho_l \rightarrow_{\mathcal{V}} g|_n$. ϕ is called the *matching \mathcal{V} -homomorphism* of the rule application, and $g|_n$ is called a ρ -*redex*. Next, we define the *result* of the application of the rule ρ to g at n using the \mathcal{V} -homomorphism ϕ . This is done by constructing the intermediate graphs g_1 and g_2 , and the final result g_3 .

- (i) The graph g_1 is obtained from g by adding the part of ρ that is not contained in its left-hand side:

$$\begin{aligned} N^{g_1} &= N^g \uplus (N^\rho \setminus N^{\rho_l}) \\ \text{lab}^{g_1}(m) &= \begin{cases} \text{lab}^g(m) & \text{if } m \in N^g \\ \text{lab}^\rho(m) & \text{if } m \in N^\rho \setminus N^{\rho_l} \end{cases} \\ \text{suc}_i^{g_1}(m) &= \begin{cases} \text{suc}_i^g(m) & \text{if } m \in N^g \\ \text{suc}_i^\rho(m) & \text{if } m, \text{suc}_i^\rho(m) \in N^\rho \setminus N^{\rho_l} \\ \phi(\text{suc}_i^\rho(m)) & \text{if } m \in N^\rho \setminus N^{\rho_l}, \text{suc}_i^\rho(m) \in N^{\rho_l} \end{cases} \end{aligned}$$

- (ii) Let $n' = \phi(r^\rho)$ if $r^\rho \in N^{\rho_l}$ and $n' = r^\rho$ otherwise. The graph g_2 is obtained from g_1 by redirecting edges ending in n to n' :

$$N^{g_2} = N^{g_1} \quad \text{lab}^{g_2} = \text{lab}^{g_1} \quad \text{suc}_i^{g_2}(m) = \begin{cases} \text{suc}_i^{g_1}(m) & \text{if } \text{suc}_i^{g_1}(m) \neq n \\ n' & \text{if } \text{suc}_i^{g_1}(m) = n \end{cases}$$

- (iii) The term graph g_3 is obtained by setting the root node r' , which is n' if $n = r^g$, and otherwise r^g . That is, $g_3 = g_2|_{r'}$. This also means that all nodes not reachable from r' are removed.

This induces a *pre-reduction step* $\psi = (g, n, \rho, n', g_3)$ from g to g_3 , written $\psi: g \mapsto_{n, \rho, n'} g_3$. In order to indicate the underlying GRS \mathcal{R} , we also write $\psi: g \mapsto_{\mathcal{R}} g_3$.

Examples for term graph (pre-)reduction steps are shown in Figure 8. We revisit them in more detail in Example 8.9 below.

The definition of term graph rewriting in the form of pre-reduction steps is very operational in style. The result of applying a rewrite rule to a term graph is constructed in several steps by manipulating nodes and edges explicitly. While this is beneficial for implementing a rewriting system, it is problematic for reasoning on term graphs up to isomorphisms,

which is necessary for introducing notions of convergence. In our case, however, this does not cause any harm since the construction in Definition 8.3 is invariant under isomorphism:

Proposition 8.4 (pre-reduction steps). *Let $\phi: g \mapsto_{n,\rho,m} h$ be a pre-reduction step in some GRS \mathcal{R} and $\psi_1: g' \cong g$. Then there is a pre-reduction step $\phi': g' \mapsto_{n',\rho,m'} h'$ with $\psi_2: h' \cong h$ such that $\psi_2(n') = n$ and $\psi_1(m') = m$.*

Proof. Immediate from the construction in Definition 8.3. \square

The above finding justifies the following definition of reduction steps:

Definition 8.5 (reduction steps). Let $\mathcal{R} = (\Sigma, R)$ be a GRS, $\rho \in R$ and $g, h \in \mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma)$ with $n \in N^g$ and $m \in N^h$. A tuple $\phi = (g, n, \rho, m, h)$ is called a *reduction step*, written $\phi: g \rightarrow_{n,\rho,m} h$, if there is a pre-reduction step $\phi': g' \mapsto_{n',\rho,m'} h'$ with $\mathcal{C}(g') = g$, $\mathcal{C}(h') = h$, $n = \mathcal{P}_{g'}(n')$, and $m = \mathcal{P}_{h'}(m')$. Similarly to pre-reduction steps, we also write $\phi: g \rightarrow_{\mathcal{R}} h$ or simply $\phi: g \rightarrow h$ for short.

In other words, a reduction step is a canonicalised pre-reduction step.

Note that term graph rules do not provide a duplication mechanism. Each variable is allowed to occur at most once. Duplication must always be simulated by sharing. This means for example that variables that should occur on the right-hand side must share the occurrence of that variable on the left-hand side of the rule with its right-hand side. This can be seen in the term graph rules in Figure 8a. The sharing can be direct as in ρ_1 or indirect as in ρ_2 . For variables that are supposed to be duplicated on the right-hand side, for example in the term rewrite rule $h(x) \rightarrow f(h(x), h(x))$, we have to use sharing in order to represent multiple occurrences of the same variable. This representation can be seen in the corresponding term graph rules in Figure 9a.

8.2. Convergence of Transfinite Reductions. We now employ the partial order \leq_{\perp}^R and the metric \mathbf{d}_{\dagger} for the purpose of defining convergence of transfinite term graph reductions.

The notion of (transfinite) reductions carries over to GRSs straightforwardly:

Definition 8.6 (transfinite reductions). Let $\mathcal{R} = (\Sigma, R)$ be a GRS. A (transfinite) *reduction* in \mathcal{R} is a sequence $(g_i \rightarrow_{\mathcal{R}} g_{i+1})_{i < \alpha}$ of rewriting steps in \mathcal{R} .

Analogously to reductions in TRSs, we need a notion of convergence in order to define well-behaved reductions. The two modes of convergence that we introduced for this very purpose in Section 5 and Section 6 are only defined on canonical term graphs. It is therefore crucial to work on reduction steps as opposed to pre-reduction steps.

Definition 8.7 (convergence of reductions). Let $\mathcal{R} = (\Sigma, R)$ be a GRS.

- (i) Let $S = (g_i \rightarrow_{\mathcal{R}} g_{i+1})_{i < \alpha}$ be a reduction in \mathcal{R} . S is *m-continuous* in \mathcal{R} , written $S: g_0 \xrightarrow{m}_{\mathcal{R}} \dots$, if the underlying sequence of term graphs $(g_i)_{i < \hat{\alpha}}$ is continuous in \mathcal{R} , i.e. $\lim_{i \rightarrow \lambda} g_i = g_{\lambda}$ for each limit ordinal $\lambda < \alpha$. S *m-converges* to $g \in \mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma)$ in \mathcal{R} , written $S: g_0 \xrightarrow{m}_{\mathcal{R}} g$, if it is *m-continuous* and $\lim_{i \rightarrow \hat{\alpha}} g_i = g$.
- (ii) Let \mathcal{R}_{\perp} be the GRS (Σ_{\perp}, R) over the extended signature Σ_{\perp} and $S = (g_i \rightarrow_{\mathcal{R}_{\perp}} g_{i+1})_{i < \alpha}$ a reduction in \mathcal{R}_{\perp} . S is *p-continuous* in \mathcal{R} , written $S: g_0 \xrightarrow{p}_{\mathcal{R}} \dots$, if $\liminf_{i \rightarrow \lambda} g_i = g_{\lambda}$ for each limit ordinal $\lambda < \alpha$. S *p-converges* to $g \in \mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp})$ in \mathcal{R} , written $S: g_0 \xrightarrow{p}_{\mathcal{R}} g$, if it is *p-continuous* and $\liminf_{i \rightarrow \hat{\alpha}} g_i = g$.

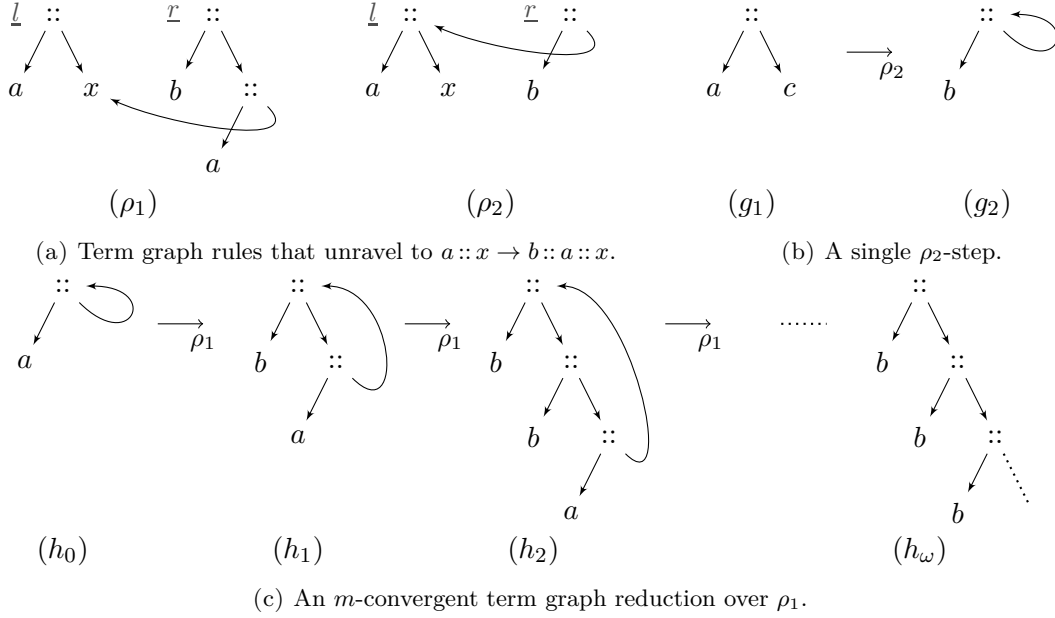


FIGURE 8. Term graph rules and their reductions.

- (iii) Let $S = (g_\iota \rightarrow_{\mathcal{R}_\perp} g_{\iota+1})_{\iota < \alpha}$ be a reduction in \mathcal{R}_\perp . The reduction S is called *p-continuous in $\mathcal{G}_C^\infty(\Sigma)$* , if it is *p-continuous* and $g_\iota \in \mathcal{G}_C^\infty(\Sigma)$ for all $\iota < \hat{\alpha}$. The reduction S is said to *p-converge in $\mathcal{G}_C^\infty(\Sigma)$* to g , if it is *p-continuous in $\mathcal{G}_C^\infty(\Sigma)$* and *p-converges* to $g \in \mathcal{G}_C^\infty(\Sigma)$.

Note that, analogously to *p-convergence* on terms, we extended the signature of \mathcal{R} to Σ_\perp for the definition of *p-convergence*. Like for terms, this approach serves two purposes. First, by considering the extended signature Σ_\perp , we allow any partial term graph to appear in a reduction as opposed to only total ones. Consequently, we have the whole complete semilattice $(\mathcal{G}_C^\infty(\Sigma_\perp), \leq_\perp^R)$ at our disposal, which means that *p-continuity* coincides with *p-convergence*:

Proposition 8.8. *In a GRS, every p-continuous reduction is p-convergent.*

Proof. Follows immediately from Corollary 5.16. □

The second reasons for the extension to \mathcal{R}_\perp is that by not presupposing that the system's signature Σ already contains a designated \perp -symbol, we rule out the possibility that this \perp symbol occurs in one of the rules of the system. Consequently, any \perp symbol present in the final term graph of a reduction is either due to the initial term graph or the convergence behaviour. This is crucial for establishing a correspondence result between *m*- and *p*-convergence in the vein of Theorem 3.3.

Example 8.9. Consider the term graph rule ρ_1 in Figure 8a, which unravels to the term rule $a :: x \rightarrow b :: a :: x$ from Example 3.1. Starting with the term tree $a :: c$, depicted as g_1 in Figure 8b, we obtain the same transfinite reduction as in Example 3.1:

$$S: a :: c \rightarrow_{\rho_1} b :: a :: c \rightarrow_{\rho_1} b :: b :: a :: c \rightarrow_{\rho_1} \dots$$

Since the modes of convergence of both the partial order \leq_1^R and the metric \mathbf{d}_\dagger coincide with the corresponding modes of convergence on terms (cf. Proposition 5.19 respectively Proposition 6.16), we know that, for reductions consisting only of term trees, both m - and p -convergence in GRSs coincide with the corresponding notions of convergence in TRSs. This observation applies to the reduction S above. Hence, also in this setting of term graph rewriting, S both m - and p -converges to the term tree h_ω shown in Figure 8c. Similarly, we can reproduce the p -converging but not m -converging reduction T from Example 3.2.

Notice that h_ω is a rational term tree as it can be obtained by unravelling the finite term graph g_2 depicted in Figure 8b. In fact, if we use the rule ρ_2 , which unravels to the term rule $a :: x \rightarrow b :: a :: x$ as well, we can immediately rewrite g_1 to g_2 . In ρ_2 , not only the variable x is shared but the whole left-hand side of the rule. This causes each redex of ρ_2 to be *captured* by the right-hand side [15].

Figure 8c indicates a transfinite reduction starting with a cyclic term graph h_0 that unravels to the rational term $t = a :: a :: a :: \dots$. This reduction both m - and p -converges to the rational term tree h_ω as well. Again, by using ρ_2 instead of ρ_1 , we can rewrite h_0 to the cyclic term graph g_2 in one step.

For more detailed explanations of the underlying modes of partial order and metric convergence for the reductions above, revisit Example 5.17 and Example 6.14, respectively.

The following theorem shows that the total fragment of p -converging reductions is in fact equivalent to m -converging reductions:

Theorem 8.10 (p -convergence in $\mathcal{G}_C^\infty(\Sigma) = m$ -convergence). *For every reduction S in a GRS the following equivalences hold:*

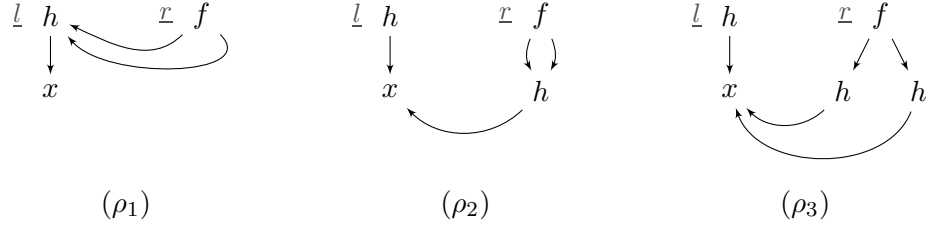
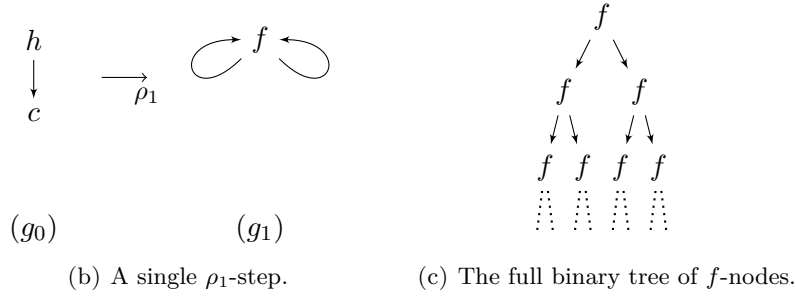
- (i) $S: g \xrightarrow{\mathcal{R}} h$ in $\mathcal{G}_C^\infty(\Sigma)$ iff $S: g \xrightarrow{m} h$
- (ii) $S: g \xrightarrow{\mathcal{R}} \dots$ in $\mathcal{G}_C^\infty(\Sigma)$ iff $S: g \xrightarrow{m} \dots$

Proof. We only show (i) since (ii) follows similarly.

Let $S = (g_i \rightarrow_{\mathcal{R}} g_{i+1})_{i < \alpha}$. For the “only if” direction assume $S: g \xrightarrow{\mathcal{R}} h$ is p -converging in $\mathcal{G}_C^\infty(\Sigma)$. Since S p -converges in $\mathcal{G}_C^\infty(\Sigma)$, it is a reduction in \mathcal{R} . The p -convergence of S implies that $\liminf_{i \rightarrow \lambda} g_i = g_\lambda$ for each limit ordinal $\lambda < \alpha$. Since each g_i is total, we have, according to Proposition 7.5, that $\lim_{i < \lambda} g_i = \liminf_{i \rightarrow \lambda} g_i = g_\lambda$ for each limit ordinal $\lambda < \alpha$. Hence $(g_i)_{i < \alpha}$ is continuous in the metric space. Likewise, we also have $\lim_{i < \alpha} g_i = \liminf_{i \rightarrow \alpha} g_i = h$. That is, S m -converges to h . For the “if” direction assume $S: g \xrightarrow{m} h$. Since $(g_i)_{i < \alpha}$ is continuous, we have that $\lim_{i < \lambda} g_i = g_\lambda$ for each limit ordinal $\lambda < \alpha$. According to Proposition 7.3, we then have that $\liminf_{i \rightarrow \lambda} g_i = g_\lambda$ for each limit ordinal $\lambda < \alpha$. Likewise we also have $\liminf_{i \rightarrow \alpha} g_i = \lim_{i < \alpha} g_i = h$. Hence, S is p -converging to h . Since S is m -converging it is by definition also in $\mathcal{G}_C^\infty(\Sigma)$. \square

Example 8.11. In order to represent term rewrite rules that are not right-linear, i.e. which have multiple occurrences of the same variable on the right-hand side, we have to use sharing to represent the occurrences of a variable by a single node. Consider the term rewrite rule $h(x) \rightarrow f(h(x), h(x))$ that duplicates the variable x on the right-hand side. Note that by repeatedly applying this term rewrite rule starting with term $h(c)$, we obtain a reduction that m -converges to the full binary tree depicted in Figure 9c.

Figure 9a shows three different ways of representing the term rewrite rule $h(x) \rightarrow f(h(x), h(x))$ as a term graph rule. The rule ρ_3 has the lowest degree of sharing since it shares the variable node directly; ρ_1 has the highest degree of sharing as it shares its complete left-hand side with its right-hand side; ρ_2 lies in between the two.


 (a) Term graph rules that unravel to $h(x) \rightarrow f(h(x), h(x))$.

 (b) A single ρ_1 -step.

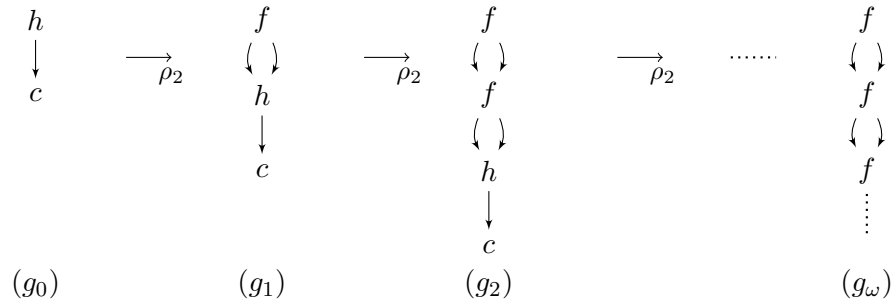
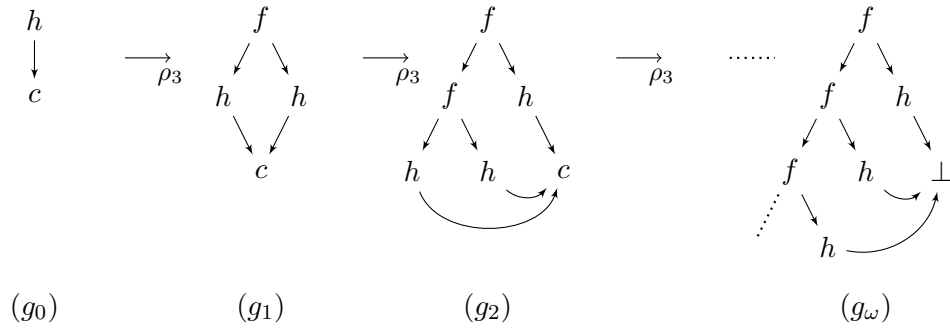
 (c) The full binary tree of f -nodes.

 (d) An m -convergent term graph reduction over ρ_2 .

 (e) A p -convergent term graph reduction over ρ_3 .

FIGURE 9. Term graph rules for duplicating term rewrite rules.

We have observed in Figure 8a before that, by sharing the complete left-hand side with the right-hand side, the redex gets captured by the right-hand side upon applying the rule to a term graph. This can be seen again in Figure 9b. By applying ρ_1 to the term tree $h(c)$ once, we immediately obtain the cyclic term graph g_1 , which unravels to the full binary tree from Figure 9c.

With the rule ρ_2 , we have to go through an m -convergent reduction of length ω , depicted in Figure 9d, in order to obtain the desired term graph normal form that then unravels to the full binary tree as well.

The same can also be achieved via the rule ρ_3 : Starting from $h(c)$ we can construct a reduction that m -converges directly to the full binary tree in Figure 9c. However, we may also form the reduction shown in Figure 9e in which we always contract the leftmost redex. As we can see in the picture, this means that the c -node remains constantly at depth 2 while still reachable from any other node. As we explained in Example 6.14, this means that the reduction does not m -converge. On the other hand, as described in Example 5.17 the reduction p -converges to the partial term graph g_ω . In fact, from this term graph g_ω we can then construct a reduction that p -converges to the full binary tree.

9. TERM GRAPH REWRITING VS. TERM REWRITING

In order to assess the value of the modes of convergence on term graphs that we introduced in this paper, we need to compare them to the well-established counterparts on terms. We have already observed that, if restricted to term trees, both the partial order \leq_\perp^R and the metric \mathbf{d}_\dagger on term graphs coincide with corresponding structures \leq_\perp and \mathbf{d} on terms, cf. Corollary 5.11 and Corollary 6.15, respectively. The same holds for the modes of convergence derived from these structures, cf. Proposition 5.19 and Proposition 6.16.

9.1. Soundness & Completeness of Infinitary Term Graph Rewriting. Ideally, we would like to see a strong connection between converging reductions in a GRS \mathcal{R} and converging reductions in the TRS $\mathcal{U}(\mathcal{R})$ that is its unravelling. For example, for m -convergence we want to see that $g \xrightarrow{\mathcal{R}} h$ implies $\mathcal{U}(g) \xrightarrow{\mathcal{U}(\mathcal{R})} \mathcal{U}(h)$ – i.e. soundness – and vice versa that $\mathcal{U}(g) \xrightarrow{\mathcal{U}(\mathcal{R})} \mathcal{U}(h)$ implies $g \xrightarrow{\mathcal{R}} h$ – i.e. completeness.

Completeness is already an issue for finitary rewriting [11]: a single term graph redex may corresponds to several term redexes due to sharing. Hence, contracting a term graph redex may correspond to several term rewriting steps. For example, given a rewrite rule $a \rightarrow b$, we can rewrite $f(a, a)$ to $f(a, b)$, whereas we can rewrite

$$\begin{array}{c} f \\ \left(\downarrow \right) \\ a \end{array} \quad \text{only to} \quad \begin{array}{c} f \\ \left(\downarrow \right) \\ b \end{array}$$

which corresponds to a term reduction $f(a, a) \rightarrow f(b, b)$. That is, in the term graph we cannot choose which of the two term redexes to contract as they are represented by the same term graph redex.

Note that there are techniques to circumvent this problem by incorporating reduction steps that copy nodes in order to reduce the sharing in a term graph [32]. In this paper, however, we are only concerned with pure term graph rewriting steps derived from rewrite rules.

In the context of weak convergence, also soundness becomes an issue. The underlying reason for this issue is similar to the phenomenon explained above: a single term graph rewrite step may represent several term rewriting steps, i.e. $g \rightarrow_{\mathcal{R}} h$ implies $\mathcal{U}(g) \rightarrow_{\mathcal{U}(\mathcal{R})}^+ \mathcal{U}(h)$.² When we have a converging term graph reduction $(g_i \rightarrow g_{i+1})_{i < \alpha}$, we know that the underlying sequence of term graphs (g_i) converges. However, the corresponding term reduction does not necessarily produce the sequence $(\mathcal{U}(g_i))$ but may intersperse the sequence $(\mathcal{U}(g_i))$ with additional intermediate terms, which might change the convergence behaviour.

A similar phenomenon is known in infinitary lambda calculus[25]: while one can simulate certain term rewriting systems with lambda terms, this simulation may fail for infinitary rewriting since a single term rewriting step may require several β -reduction steps. The problem that arises in this setting is that the intermediate terms that are introduced in the lambda reduction may cause the convergence to break.

The same can, in principle, also occur when simulating a term graph reduction by a term reduction. Since a single term graph rewrite step may require several term rewrite steps, we may introduce intermediate terms into the reduction that do not directly correspond to the term graphs in the graph reduction.

9.2. Preservation of Convergence under Unravelling. Due to the abovementioned difficulties, we restrict ourselves in this paper to the soundness of the modes of convergence alone. By soundness in this setting we mean that whenever we have a sequence $(g_i)_{i < \alpha}$ of term graphs converging to g , then the sequence $(\mathcal{U}(g_i))_{i < \alpha}$ converges to $\mathcal{U}(g)$. That is, convergence is preserved under unravelling. Since the metric \mathbf{d}_{\dagger} on term graphs generalises the metric \mathbf{d} on terms, cf. Corollary 6.15, it does not matter whether we consider the convergence of $(\mathcal{U}(g_i))_{i < \alpha}$ in the metric space $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma), \mathbf{d}_{\dagger})$ or $(\mathcal{T}^{\infty}(\Sigma), \mathbf{d})$, according to Proposition 6.16. The same also holds for the limit inferior in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$ and $(\mathcal{T}^{\infty}(\Sigma_{\perp}), \leq_{\perp})$, due to Corollary 5.11 and Proposition 5.19.

The cornerstone of the investigation of the unravelling of term graphs is the following simple characterisation of unravelling in terms of labelled quotient trees:

Proposition 9.1. *The unravelling $\mathcal{U}(g)$ of a term graph $g \in \mathcal{G}^{\infty}(\Sigma)$ is given by the labelled quotient tree $(\mathcal{P}(g), g(\cdot), \mathcal{I}_{\mathcal{P}(g)})$.*

Proof. Since $\mathcal{I}_{\mathcal{P}(g)}$ is a subrelation of \sim_g , we know that $(\mathcal{P}(g), g(\cdot), \mathcal{I}_{\mathcal{P}(g)})$ is a labelled quotient tree and thus uniquely determines a term tree t . By Lemma 4.19, there is a homomorphism from t to g . Hence, $\mathcal{U}(g) = t$. \square

Employing the above characterisation, we can easily see that the relation \leq_{\perp}^R is preserved under unravelling:

Proposition 9.2. *Given $g, h \in \mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp})$, we have that $g \leq_{\perp}^R h$ implies $\mathcal{U}(g) \leq_{\perp}^R \mathcal{U}(h)$.*

Proof. Immediate consequence of Corollary 5.10 and Proposition 9.1. \square

²If the term graph g is cyclic, the corresponding term reduction may even be infinite.

Likewise, also least upper bounds of \leq_{\perp}^R are preserved:

Proposition 9.3 (preservation of lubs under unravelling). *Given a directed set G in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$, also $\{\mathcal{U}(g) \mid g \in G\}$ is directed and $\mathcal{U}(\bigsqcup_{g \in G} g) = \bigsqcup_{g \in G} \mathcal{U}(g)$.*

Proof. The fact that $\{\mathcal{U}(g) \mid g \in G\}$ is directed in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$ follows from Proposition 9.2. The equality follows from the characterisation of the lub in Theorem 5.12 and from Proposition 9.1. \square

For greatest lower bounds of \leq_{\perp}^R , the situation is more complicated as we have to consider arbitrary non-empty sets of term graphs instead of only directed sets.

We start with the characterisation of glbs in the partially ordered set $(\mathcal{T}^{\infty}(\Sigma_{\perp}), \leq_{\perp})$ of terms. Since this partially ordered set forms a complete semi-lattice, we know that it admits glbs of arbitrary non-empty sets. The following lemma characterises these glbs:

Lemma 9.4 (glb on terms). *The glb $\prod T$ of a non-empty set T in $(\mathcal{T}^{\infty}(\Sigma_{\perp}), \leq_{\perp})$ is given by the labelled quotient tree (P, l, \mathcal{I}_P) where*

$$P = \left\{ \pi \in \bigcap_{t \in T} \mathcal{P}(t) \mid \forall \pi' < \pi \exists f \in \Sigma_{\perp} \forall t \in T : t(\pi') = f \right\}$$

$$l(\pi) = \begin{cases} f & \text{if } \forall t \in T : f = t(\pi) \\ \perp & \text{otherwise} \end{cases}$$

Proof. Special case of Proposition 5.9 in [10]. \square

By combining the above characterisation with the characterisation of unravelled term graphs, we obtain the following:

Corollary 9.5. *Given a non-empty set G in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$, the glb $\prod_{g \in G} \mathcal{U}(g)$ is given by the labelled quotient tree (P, l, \mathcal{I}_P) where*

$$P = \left\{ \pi \in \bigcap_{g \in G} \mathcal{P}(g) \mid \forall \pi' < \pi \exists f \in \Sigma_{\perp} \forall g \in G : g(\pi') = f \right\}$$

$$l(\pi) = \begin{cases} f & \text{if } \forall g \in G : f = g(\pi) \\ \perp & \text{otherwise} \end{cases}$$

Proof. Follows immediately from Lemma 9.4 and Proposition 9.1. \square

Before we deal with the preservation of glbs under unravelling, we need the following property that relates the unravelling of a glb to the original term graphs:

Lemma 9.6 (unravelling of a glb). *For each non-empty set G in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$, the term $t = \mathcal{U}(\prod G)$ satisfies the following for all $g \in G$ and $\pi \in \mathcal{P}(t)$:*

$$(i) \ \pi \in \mathcal{P}(g) \qquad (ii) \ t(\pi) \in \Sigma \implies t(\pi) = g(\pi)$$

Proof. Let $g \in G$, $\pi \in \mathcal{P}(t)$, and $h = \prod G$. Then $\pi \in \mathcal{P}(h)$ and $h(\pi) = t(\pi)$ according to Proposition 9.1. Since $h \leq_{\perp}^R g$, we may apply Corollary 5.10 to obtain (i) that $\pi \in \mathcal{P}(g)$ and (ii) that $t(\pi) = g(\pi)$ whenever $t(\pi) \in \Sigma$. \square

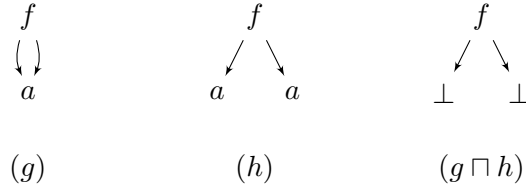


FIGURE 10. Failure of preservation of glbs under unravelling.

Proposition 9.7 (weak preservation of glbs under unravelling). *If G is a non-empty set in $(\mathcal{G}_C^\infty(\Sigma_\perp), \leq_\perp^R)$, then $\mathcal{U}\left(\bigcap_{g \in G} g\right) \leq_\perp^R \bigcap_{g \in G} \mathcal{U}(g)$.*

Proof. Let $s = \mathcal{U}\left(\bigcap_{g \in G} g\right)$ and $t = \bigcap_{g \in G} \mathcal{U}(g)$. Since both s and t are terms, we can use the characterisation of \leq_\perp instead of \leq_\perp^R . That is, we will show that for each $\pi \in \mathcal{P}(s)$, we have that $\pi \in \mathcal{P}(t)$ and that $t(\pi) = s(\pi)$ whenever $s(\pi) \in \Sigma$.

If $\pi \in \mathcal{P}(s)$, then $\pi' \in \mathcal{P}(s)$ for all $\pi' < \pi$. Since $s(\pi')$ cannot be a nullary symbol if $\pi' < \pi$, we know that $s(\pi') \neq \perp$. Hence, we can apply Lemma 9.6 in order to obtain for all $g \in G$ that $\pi \in \mathcal{P}(g)$ and that $s(\pi') = g(\pi')$ for all $\pi' < \pi$. According to Corollary 9.5, this means that $\pi \in \mathcal{P}(t)$. In order to show the second part, assume that $s(\pi) \in \Sigma$. Then, by Lemma 9.6, $g(\pi) = s(\pi)$ for all $g \in G$, which, according to Corollary 9.5, implies that $t(\pi) = s(\pi)$. \square

In general, glbs are not fully preserved under unravelling as the following example shows:

Example 9.8. Consider the term graphs g and h in Figure 10. The only difference between the two term graphs is the sharing of the arguments of the root node. Due to this difference in sharing, the glb $g \sqcap h$ of the two term graphs is a proper partial term graph as depicted in Figure 10. On the other hand, since the unravelling of the two term graphs coincides, viz. $\mathcal{U}(g) = \mathcal{U}(h) = h$, we have that $\mathcal{U}(g) \sqcap \mathcal{U}(h) = h$. In particular, we have the strict inequality $\mathcal{U}(g \sqcap h) <_\perp^R \mathcal{U}(g) \sqcap \mathcal{U}(h)$.

Unfortunately, this also means that the limit inferior is only weakly preserved under unravelling as well:

Theorem 9.9. *For every sequence $(g_\iota)_{\iota < \alpha}$ in $(\mathcal{G}_C^\infty(\Sigma_\perp), \leq_\perp^R)$, we have that*

$$\mathcal{U}\left(\liminf_{\iota \rightarrow \alpha} g_\iota\right) \leq_\perp^R \liminf_{\iota \rightarrow \alpha} \mathcal{U}(g_\iota).$$

Proof. This follows from Proposition 9.7 and Proposition 9.3. \square

Again, we can construct a counterexample that shows that the converse inequality does not hold in general:

Example 9.10. Let $(g_\iota)_{\iota < \omega}$ be the sequence alternating between g and h from Figure 10, i.e. $g_{2\iota} = g$ and $g_{2\iota+1} = h$ for all $\iota < \omega$. Then $\bigcap_{\alpha \leq \iota < \omega} g_\iota = g \sqcap h$ for each $\alpha < \omega$ and, consequently, $\liminf_{\iota \rightarrow \omega} g_\iota = g \sqcap h$. As we have seen in Example 9.8, $g \sqcap h$ is the proper partial term graph depicted in Figure 10. On the other hand, since $\mathcal{U}(g) = \mathcal{U}(h) = h$, we have that $\liminf_{\iota \rightarrow \omega} \mathcal{U}(g_\iota) = h$. In particular, we have the strict inequality $\mathcal{U}(\liminf_{\iota \rightarrow \omega} g_\iota) <_\perp^R \liminf_{\iota \rightarrow \omega} \mathcal{U}(g_\iota)$.

Moreover, we cannot expect that any other partial order with properties comparable to those of \leq_{\perp}^R fully preserves the limit inferior under unravelling.

The example above shows that any partial order \leq on partial term graphs whose limit inferior is preserved under unravelling must also satisfy either $g \leq h$ or $h \leq g$ for the term graphs in Figure 10. That is, such a partial order has to give up the property that total term graphs are maximal, cf. Proposition 5.20. This observation is independent of whether this partial order specialises to \leq_{\perp} on terms.

The sacrifice for full preservation under unravelling goes even further. If a partial order \leq on partial term graphs satisfies preservation of its limit inferior under unravelling, the limit inferior $\liminf_{i \rightarrow \omega} g_i$ of the sequence $(g_i)_{i < \omega}$ from Example 9.10 has to unravel to h , a total term. That is, $\liminf_{i \rightarrow \omega} g_i$ has to be a total term graph. On the other hand, there is no metric – or any Hausdorff topology for that matter – for which $(g_i)_{i < \omega}$ converges at all because $(g_i)_{i < \omega}$ alternates between two distinct term graphs. In other words, the correspondence between m - and p -convergence, which we have for \leq_{\perp}^R as stated in Theorem 8.10, cannot be satisfied for such a partial order \leq , regardless of the metric on term graphs.

The simple partial order \leq_{\perp}^S , which we briefly discussed in comparison to the rigid partial order \leq_{\perp}^R in Section 5, takes the other side of the trade-off illustrated above: it satisfies $g \leq_{\perp}^S h$ and the preservation of the limit inferior under unravelling but sacrifices the correspondence between total term graphs and maximality as well as the correspondence between m - and p -convergence [10].

Using the correspondence between the limit inferior in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$ and the limit in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma), \mathbf{d}_{\dagger})$, we can derive full preservation of limits under unravelling:

Theorem 9.11. *For every convergent sequence $(g_i)_{i < \alpha}$ in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma), \mathbf{d}_{\dagger})$, also $(\mathcal{U}(g_i))_{i < \alpha}$ is convergent and*

$$\mathcal{U}\left(\lim_{i \rightarrow \alpha} g_i\right) = \lim_{i \rightarrow \alpha} \mathcal{U}(g_i).$$

Proof. We prove the equality as follows:

$$\mathcal{U}\left(\lim_{i \rightarrow \alpha} g_i\right) \stackrel{(1)}{=} \mathcal{U}\left(\liminf_{i \rightarrow \alpha} g_i\right) \stackrel{(2)}{=} \liminf_{i \rightarrow \alpha} \mathcal{U}(g_i) \stackrel{(3)}{=} \lim_{i \rightarrow \alpha} \mathcal{U}(g_i)$$

- (1) Since $(g_i)_{i < \alpha}$ is convergent, and thus Cauchy, we can apply Proposition 7.3 to obtain that $\lim_{i \rightarrow \alpha} g_i = \liminf_{i \rightarrow \alpha} g_i$.
- (2) Since $\liminf_{i \rightarrow \alpha} g_i$ is total, so is $\mathcal{U}(\liminf_{i \rightarrow \alpha} g_i)$. By Proposition 5.20, this means that $\mathcal{U}(\liminf_{i \rightarrow \alpha} g_i)$ is maximal w.r.t. \leq_{\perp}^R . Consequently, the inequality $\mathcal{U}(\liminf_{i \rightarrow \alpha} g_i) \leq_{\perp}^R \liminf_{i \rightarrow \alpha} \mathcal{U}(g_i)$ due to Theorem 9.9 yields (2).
- (3) This equality follows from Proposition 7.5 and the totality of $\liminf_{i \rightarrow \alpha} \mathcal{U}(g_i)$. \square

9.3. Finite Representations of Transfinite Term Reductions. One of the motivations for considering modes of convergence on term graphs in the first place is the study of finite representation of transfinite term reductions as finite term graph reductions. Since both the metric \mathbf{d}_{\dagger} and the partial order \leq_{\perp}^R specialise to the corresponding structures on terms, we can use both the metric space $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma), \mathbf{d}_{\dagger})$ and the partially ordered set $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$ to move seamlessly from terms to term graphs and vice versa.

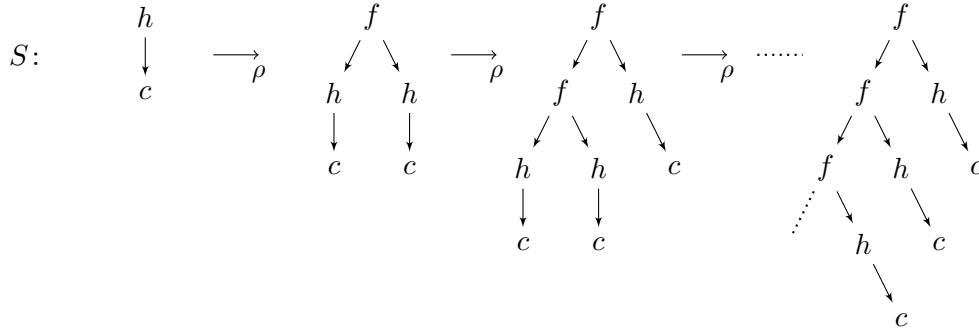
For instance, Example 8.11 illustrates reductions that perform essentially the same computations, however, at different levels of sharing / parallelism. This includes the complete lack of sharing as well, i.e. term rewriting. For each of the cases we can use the partially

ordered set $(\mathcal{G}_C^\infty(\Sigma_\perp), \leq_\perp^R)$ and the metric space $(\mathcal{G}_C^\infty(\Sigma), \mathbf{d}_\dagger)$ as a unifying framework to determine the convergence behaviour.

In order to use the partial order \leq_\perp^R and the metric \mathbf{d}_\dagger as a tool to study finite representability of infinite term reductions as finite term graph reductions there still is some work to be done, though.

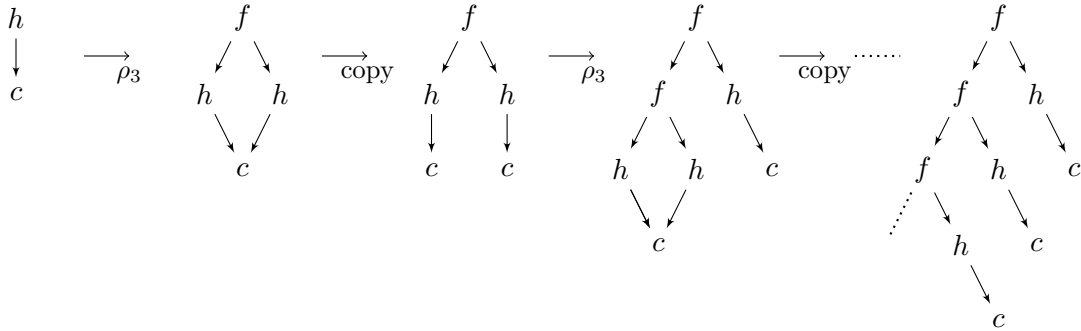
First and foremost, we need a unifying framework for performing both term and term graph rewriting. A straightforward approach to achieve this, is to include copying steps in term graph reductions that allow us to revert the sharing produced by applying term graph rules [32]. For example while the rule ρ_3 from Figure 9a is the term graph rule with the least sharing that unravels to $\rho: h(x) \rightarrow f(h(x), h(x))$, it still has some sharing in order to represent the duplication of the variable x on the right-hand side.

The result of this sharing is seen in Figure 9e, which shows that even if we start with a term tree, the rule ρ_3 turns it into a proper term graph. Consequently it is slightly different from the corresponding term reduction



In fact, while the above term reduction S is m -convergent, the term graph reduction via ρ_3 , depicted in Figure 9e, is not.

However, by interspersing the term graph reduction with reduction steps that copy nodes – and in general sub-term graphs – we instead obtain the following term graph reduction:



This reduction simulates the corresponding term reduction S more closely and in fact both reductions m -converge to the same term. Nevertheless, this approach creates the same issue that we have already noted for soundness: the additional term graphs that get interspersed with the original term reduction may affect the convergence behaviour.

The second ingredient that we need in order to study the finite representability of transfinite term reductions is a compression property [27] for transfinite term graph reductions

that allows us to compress a transfinite term graph reduction that ends in a finite term graph to a term graph reduction of finite length.

Unfortunately, experience from infinitary term rewriting already shows us that a general compression property – allowing any reduction to be compressed to length at most ω – is not possible for weak convergence [24]. However, the more restrictive version of the compression property that we need, viz. that reductions ending in a finite term graph may be compressed to finite length, does hold for weakly m -converging term reductions [28] and there is hope that this carries over to the term graph rewriting setting.

10. CONCLUSIONS & FUTURE WORK

With the goal of generalising infinitary term rewriting to term graphs, we have presented two different modes of convergence for an infinitary calculus of term graph rewriting. The success of this generalisation effort was demonstrated by a number of results. Many of the properties of the modes of convergence on terms have been maintained in this transition to term graphs. First and foremost, this includes the intrinsic completeness properties of the underlying structures, i.e. the metric space is still complete and the partially ordered set still forms a complete semilattice. Moreover, we were also able to maintain the correspondence between p - and m -convergence as well as the intuition of the partial order to capture a notion of information preservation.

An important check for the appropriateness of the modes of convergence on term graphs is their relation to the corresponding modes of convergence on terms. For both the partial order and the metric approach, we have that convergence on term graphs is a conservative extension of the convergence on terms. Conversely, convergence on term graphs carries over to convergence on terms via the unravelling mapping. Unfortunately, this preservation of convergence under unravelling is only weak in the case of the partial order setting; cf. Theorem 9.9. However, as we have explained in Section 9.2, this phenomenon is an unavoidable side effect of the generalisation to term graphs unless other important properties are sacrificed. Fortunately, this phenomenon vanishes in the metric setting and we in fact obtain full preservation of limits under unravelling; cf. Theorem 9.11.

As a result, we have obtained two modes of convergence, which allow us to combine both infinitary term rewriting and term graph rewriting within one theoretical framework. Our motivation for this effort is derived from studying *lazy evaluation* and the correspondence between infinitary term rewriting and finitary term graph rewriting. For both applications, we still require more understanding of the matter, though: for the former, we still lack at least a treatment of higher-order rewriting whereas we are much closer to the latter. We have discussed issues concerning the correspondence between infinitary term rewriting and finitary term graph rewriting in detail in Section 9.3: while the unified modes of convergence are already helpful for studying infinitary rewriting with a varying degree of sharing, we identified two shortcomings that have to be addressed, viz. the lack of a unifying notion of rewriting for terms and term graphs and a compression property for transfinite term graph reductions.

Apart from the abovementioned issues, future work should also be concerned with establishing a stronger correspondence between infinitary term rewriting and infinitary term graph rewriting beyond the preservation of limits under unravellings, which we showed in this paper. Despite the difficulties that we encountered in Section 9.1, we think that obtaining such results is possible. However, a more promising way of approaching this

issue is to restrict the notion of convergence to strong convergence as we know it from infinitary term rewriting [27]. Such a stricter notion of convergence takes the location of a reduction step into consideration and, thus, provides a closer correspondence between term graph reductions and their term rewriting counterparts. Indeed, this technique has been applied successfully to convergence on term graphs based on the simple partial order \leq_{\perp}^S , which we briefly discussed in comparison to the rigid partial order \leq_{\perp}^R in Section 5, and a corresponding metric [10].

ACKNOWLEDGEMENT

The author would like to thank the anonymous referees of RTA 2011 as well as the referees for the special issue of LMCS whose comments greatly helped to improve the presentation of the material.

REFERENCES

- [1] Zena Ariola and Stefan Blom. Skew and ω -Skew Confluence and Abstract Böhm Semantics. In Aart Middeldorp, Vincent van Oostrom, Femke van Raamsdonk, and Roel de Vrijer, editors, *Processes, Terms and Cycles: Steps on the Road to Infinity*, volume 3838 of *Lecture Notes in Computer Science*, pages 368–403. Springer Berlin / Heidelberg, 2005.
- [2] Zena M Ariola and Stefan Blom. Skew confluence and the lambda calculus with letrec. *Ann. Pure Appl. Logic*, 117(1-3):95–168, 2002.
- [3] Zena M Ariola and Jan Willem Klop. Lambda Calculus with Explicit Recursion. *Inf. Comput.*, 139(2):154–233, 1997.
- [4] André Arnold and Maurice Nivat. The metric space of infinite trees. Algebraic and topological properties. *Fundam. Inf.*, 3(4):445–476, 1980.
- [5] Patrick Bahr. Infinitary Rewriting - Theory and Applications. Master’s thesis, Vienna University of Technology, Vienna, 2009.
- [6] Patrick Bahr. Abstract Models of Transfinite Reductions. In Christopher Lynch, editor, *Proceedings of the 21st International Conference on Rewriting Techniques and Applications*, volume 6 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49–66, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [7] Patrick Bahr. Partial Order Infinitary Term Rewriting and Böhm Trees. In Christopher Lynch, editor, *Proceedings of the 21st International Conference on Rewriting Techniques and Applications*, volume 6 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 67–84, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [8] Patrick Bahr. Modes of Convergence for Term Graph Rewriting. In Manfred Schmidt-Schauß, editor, *22nd International Conference on Rewriting Techniques and Applications*, volume 10 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 139–154, Dagstuhl, Germany, 2011. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [9] Patrick Bahr. Convergence in infinitary term graph rewriting systems is simple. Unpublished manuscript, available from the author’s web site, 2012.
- [10] Patrick Bahr. Infinitary term graph rewriting is simple, sound and complete. RTA 2012, to appear, 2012.
- [11] Henk P Barendregt, Marko C J D van Eekelen, John R W Glauert, Richard Kennaway, Marinus J Plasmeijer, and M Ronan Sleep. Term graph rewriting. In Philip C Treleaven de Bakker A. J. Nijman, editor, *Parallel Architectures and Languages Europe, Volume II: Parallel Languages*, volume 259 of *Lecture Notes in Computer Science*, pages 141–158. Springer Berlin / Heidelberg, 1987.
- [12] Erik Barendsen. Term Graph Rewriting. In Terese, editor, *Term Rewriting Systems*, chapter 13, pages 712–743. Cambridge University Press, 2003.
- [13] Stefan Blom. An Approximation Based Approach to Infinitary Lambda Calculi. In Vincent van Oostrom, editor, *Rewriting Techniques and Applications*, volume 3091 of *Lecture Notes in Computer Science*, pages 221–232. Springer Berlin / Heidelberg, 2004.

- [14] Nachum Dershowitz, Stéphane Kaplan, and David A Plaisted. Rewrite, rewrite, rewrite, rewrite, rewrite, ... *Theor. Comput. Sci.*, 83(1):71–96, 1991.
- [15] William M Farmer and Ronald J Watro. Redex capturing in term graph rewriting. *Internat. J. Found. Comput. Sci.*, 1:369–386, 1990.
- [16] Joseph A Goguen, James W Thatcher, Eric G Wagner, and Jesse B Wright. Initial Algebra Semantics and Continuous Algebras. *J. ACM*, 24(1):68–95, 1977.
- [17] Peter Henderson and James H Morris Jr. A lazy evaluator. In *Proceedings of the 3rd ACM SIGACT-SIGPLAN symposium on Principles on programming languages*, pages 95–103, New York, NY, USA, 1976. ACM.
- [18] John Hughes. Why Functional Programming Matters. *Comput. J.*, 32(2):98–107, 1989.
- [19] Gilles Kahn and Gordon D Plotkin. Concrete domains. *Theor. Comput. Sci.*, 121(1-2):187–277, 1993.
- [20] Stefan Kahrs. Infinitary rewriting: meta-theory and convergence. *Acta Inform.*, 44(2):91–121, May 2007.
- [21] John L Kelley. *General Topology*, volume 27 of *Graduate Texts in Mathematics*. Springer-Verlag, 1955.
- [22] Richard Kennaway. On transfinite abstract reduction systems. Technical report, CWI (Centre for Mathematics and Computer Science), Amsterdam, 1992.
- [23] Richard Kennaway. Infinitary Rewriting and Cyclic Graphs. *Electron. Notes Theor. Comput. Sci.*, 2:153–166, 1995.
- [24] Richard Kennaway and Fer-Jan de Vries. Infinitary Rewriting. In Terese, editor, *Term Rewriting Systems*, chapter 12, pages 668–711. Cambridge University Press, 1st edition, 2003.
- [25] Richard Kennaway, Jan Willem Klop, M R Sleep, and Fer-Jan de Vries. Infinitary lambda calculus. *Theor. Comput. Sci.*, 175(1):93–125, 1997.
- [26] Richard Kennaway, Jan Willem Klop, M Ronan Sleep, and Fer-Jan de Vries. On the adequacy of graph rewriting for simulating term rewriting. *ACM Trans. Program. Lang. Syst.*, 16(3):493–523, 1994.
- [27] Richard Kennaway, Jan Willem Klop, M Ronan Sleep, and Fer-Jan de Vries. Transfinite Reductions in Orthogonal Term Rewriting Systems. *Inf. Comput.*, 119(1):18–38, 1995.
- [28] Salvador Lucas. Transfinite Rewriting Semantics for Term Rewriting Systems. In Aart Middeldorp, editor, *Rewriting Techniques and Applications*, volume 2051 of *Lecture Notes in Computer Science*, pages 216–230. Springer Berlin / Heidelberg, 2001.
- [29] Simon Marlow. Haskell 2010 Language Report, 2010.
- [30] Simon Peyton-Jones. *The Implementation of Functional Programming Languages*. Prentice Hall, 1987.
- [31] Rinus Plasmeijer and Marko C J D van Eekelen. *Functional Programming and Parallel Graph Rewriting*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1993.
- [32] Detlef Plump. Term graph rewriting. In Hartmut Ehrig, Gregor Engels, Hans-Jörg Kreowski, and Grzegorz Rozenberg, editors, *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 2: Applications, Languages, and Tools*, pages 3–61. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1999.
- [33] Jakob Grue Simonsen. Weak Convergence and Uniform Normalization in Infinitary Rewriting. In Christopher Lynch, editor, *Proceedings of the 21st International Conference on Rewriting Techniques and Applications*, volume 6 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 311–324, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [34] Terese. *Term Rewriting Systems*. Cambridge University Press, 1st edition, 2003.

APPENDIX A. PROOF OF LEMMA 5.14

Lemma 5.14 (compatible elements have lub). *Each pair g_1, g_2 of compatible term graphs in $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$ has a least upper bound.*

Proof of Lemma 5.14. Since $\{g_1, g_2\}$ is not necessarily directed, its lub might have positions that are neither in g_1 nor in g_2 . Therefore, it is easier to employ a different construction here: Following Remark 5.13, we will use the structure $(\mathcal{G}^{\infty}(\Sigma_{\perp})/\cong, \leq_{\perp}^R)$ which is isomorphic to $(\mathcal{G}_{\mathcal{C}}^{\infty}(\Sigma_{\perp}), \leq_{\perp}^R)$. To this end, we will construct a term graph \bar{g} such that $[\bar{g}]_{\cong}$ is the lub of $\{[g_1]_{\cong}, [g_2]_{\cong}\}$. Since we assume that $\{[g_1]_{\cong}, [g_2]_{\cong}\}$ has an upper bound, say $[\hat{g}]_{\cong}$, there are two rigid \perp -homomorphisms $\phi_i: g_i \rightarrow_{\perp} \hat{g}$.

Let $g_j = (N^j, \text{suc}^j, \text{lab}^j, r^j)$, $j = 1, 2$. Since we are dealing with isomorphism classes, we can assume w.l.o.g. that the nodes in g_j are of the form n^j for $j = 1, 2$. Let $\overline{M} = N^1 \uplus N^2$ and define the relation \sim on \overline{M} as follows:

$$n^j \sim m^k \quad \text{iff} \quad \mathcal{P}_{g_j}(n^j) \cap \mathcal{P}_{g_k}(m^k) \neq \emptyset$$

\sim is clearly reflexive and symmetric. Hence, its transitive closure \sim^+ is an equivalence relation on \overline{M} . Now define the term graph $\overline{g} = (\overline{N}, \overline{\text{lab}}, \overline{\text{suc}}, \overline{r})$ as follows:

$$\begin{aligned} \overline{N} &= \overline{M} / \sim^+ & \overline{\text{lab}}(N) &= \begin{cases} f & \text{if } f \in \Sigma, \exists n^j \in N. \text{lab}^j(n^j) = f \\ \perp & \text{otherwise} \end{cases} \\ \overline{r} &= [r^1]_{\sim^+} & \overline{\text{suc}}_i(N) &= N' \quad \text{iff} \quad \exists n^j \in N. \text{suc}_i^j(n^j) \in N' \end{aligned}$$

Note that since $\langle \rangle \in \mathcal{P}_{g_1}(r^1) \cap \mathcal{P}_{g_2}(r^2)$, we also have $\overline{r} = [r^2]_{\sim^+}$.

Before we argue about the well-definedness of \overline{g} , we need to establish some auxiliary claims:

$$n^j \sim^+ m^k \implies \phi_j(n^j) = \phi_k(m^k) \quad \text{for all } n^j, m^k \in \overline{M} \quad (1)$$

$$\phi_j(n^j) = \phi_k(m^k) \implies n^j \sim m^k \quad \begin{array}{l} \text{for all } n^j, m^k \in \overline{M} \\ \text{with } \text{lab}^j(n^j), \text{lab}^k(m^k) \in \Sigma \end{array} \quad (1')$$

We show (1) by proving that $n^j \sim^p m^k$ implies $\phi_j(n^j) = \phi_k(m^k)$ by induction on $p > 0$. If $p = 1$, then $n^j \sim m^k$. Hence, $\mathcal{P}_{g_j}(n^j) \cap \mathcal{P}_{g_k}(m^k) \neq \emptyset$. Additionally, from Lemma 4.10 we obtain both $\mathcal{P}_{g_j}(n^j) \subseteq \mathcal{P}_{\overline{g}}(\phi_j(n^j))$ and $\mathcal{P}_{g_k}(m^k) \subseteq \mathcal{P}_{\overline{g}}(\phi_k(m^k))$. Consequently, we also have that $\mathcal{P}_{\overline{g}}(\phi_j(n^j)) \cap \mathcal{P}_{\overline{g}}(\phi_k(m^k)) \neq \emptyset$, i.e. $\phi_j(n^j) = \phi_k(m^k)$. If $p = q + 1 > 1$, then there is some $o^l \in \overline{M}$ with $n^j \sim o^l$ and $o^l \sim^q m^k$. Applying the induction hypothesis immediately yields $\phi_j(n^j) = \phi_l(o^l) = \phi_k(m^k)$.

For (1'), let $n^j, m^k \in \overline{M}$ with $\text{lab}^j(n^j), \text{lab}^k(m^k) \in \Sigma$ and $\phi_j(n^j) = \phi_k(m^k)$. Since ϕ_j and ϕ_k are rigid \perp -homomorphisms, we have the following equations:

$$\mathcal{P}_{g_j}^a(n^j) = \mathcal{P}_{\overline{g}}^a(\phi_j(n^j)) = \mathcal{P}_{\overline{g}}^a(\phi_k(m^k)) = \mathcal{P}_{g_k}^a(m^k).$$

Hence, $\mathcal{P}_{g_j}(n^j) \cap \mathcal{P}_{g_k}(m^k) \neq \emptyset$ and, therefore, $n^j \sim m^k$.

Next we show that $\overline{\text{lab}}$ is well-defined. To this end, let $N \in \overline{N}$ and $n^j, m^k \in N$ such that $\text{lab}^j(n^j) = f_1 \in \Sigma$ and $\text{lab}^k(m^k) = f_2 \in \Sigma$. We need to show that $f_1 = f_2$. By (1), we have that $\phi_j(n^j) = \phi_k(m^k)$. Since $f_1, f_2 \in \Sigma$, we can employ the labelling condition for ϕ_j and ϕ_k in order to obtain that

$$f_1 = \text{lab}^j(n^j) = \widehat{\text{lab}}(\phi_j(n^j)) = \widehat{\text{lab}}(\phi_k(m^k)) = \text{lab}^k(m^k) = f_2.$$

To argue that $\overline{\text{suc}}$ is well-defined, we first have to show for all $N \in \overline{N}$ that $\overline{\text{suc}}_i(N)$ is defined iff $i < \text{ar}(\overline{\text{lab}}(N))$. Suppose that $\overline{\text{suc}}_i(N)$ is defined. Then there is some $n^j \in N$ such that $\text{suc}_i^j(n^j)$ is defined. Hence, $i < \text{ar}(\text{lab}^j(n^j))$. Since then also $\text{lab}^j(n^j) \in \Sigma$, we have $\overline{\text{lab}}(N) = \text{lab}^j(n^j)$. Therefore, $i < \text{ar}(\overline{\text{lab}}(N))$. If, conversely, there is some $i \in \mathbb{N}$ with $i < \text{ar}(\overline{\text{lab}}(N))$, then we know that $\overline{\text{lab}}(N) = f \in \Sigma$. Hence, there is some $n^j \in N$ with $\text{lab}^j(n^j) = f$. Hence, $i < \text{ar}(\text{lab}^j(n^j))$ and, therefore, $\text{suc}_i^j(n^j)$ is defined. Hence, $\overline{\text{suc}}_i(N)$ is defined.

To finish the argument showing that $\overline{\text{suc}}$ is well-defined, we have to show that, for all $N, N_1, N_2 \in \overline{N}$ and $n^j, m^k \in N$ such that $\text{suc}_i^j(n^j) \in N_1$ and $\text{suc}_i^k(m^k) \in N_2$, we indeed have $N_1 = N_2$. As $n^j, m^k \in N$, we have $n^j \sim^+ m^k$ and, therefore, $\phi_j(n^j) = \phi_k(m^k)$ according

to (1). Since both $\text{suc}_i^j(n^j)$ and $\text{suc}_i^k(m^k)$ are defined, we have $\text{lab}^j(n^j), \text{lab}^k(m^k) \in \Sigma$. By (1') we then have $n^j \sim m^k$, i.e. there is some $\pi \in \mathcal{P}_{g_j}(n^j) \cap \mathcal{P}_{g_k}(m^k)$. Consequently, $\pi \cdot \langle i \rangle \in \mathcal{P}_{g_j}(\text{suc}_i^j(n^j)) \cap \mathcal{P}_{g_k}(\text{suc}_i^k(m^k))$. Hence, $\text{suc}_i^j(n^j) \sim \text{suc}_i^k(m^k)$ and, therefore, $N_1 = N_2$.

Before we begin the main argument we need establish the following auxiliary claims:

$$\mathcal{P}_{g_j}(n^j) \subseteq \mathcal{P}_{\bar{g}}([n^j]_{\sim+}) \quad \text{for all } n^j \in \bar{M} \quad (2)$$

$$\forall \pi \in \mathcal{P}_{\bar{g}}^a(N) \exists n^j \in N. \text{lab}^j(n^j) \in \Sigma, \pi \in \mathcal{P}_{g_j}^a(n^j) \quad \text{for all } N \in \bar{N} \text{ with } \overline{\text{lab}}(N) \in \Sigma \quad (3)$$

$$n^j \sim^+ m^k \implies \mathcal{P}_{g_j}^a(n^j) = \mathcal{P}_{g_k}^a(m^k) \quad \begin{array}{l} \text{for all } n^j, m^j \in \bar{M} \\ \text{with } \text{lab}^j(n^j), \text{lab}^k(m^k) \in \Sigma \end{array} \quad (4)$$

For (2), we will show that $\pi \in \mathcal{P}_{g_j}(n^j)$ implies $\pi \in \mathcal{P}_{\bar{g}}([n^j]_{\sim+})$ by induction on the length of π . The case $\pi = \langle \rangle$ is trivial. If $\pi = \pi' \cdot \langle i \rangle$, then $\pi' \cdot \langle i \rangle \in \mathcal{P}_{g_j}(n^j)$, i.e. for $m^j = \text{node}_{g_j}(\pi')$, we have $\text{suc}_i^j(m^j) = n^j$. Employing the induction hypothesis, we obtain $\pi' \in \mathcal{P}_{\bar{g}}([m^j]_{\sim+})$. Additionally, according to the construction of \bar{g} , we have $\overline{\text{suc}}_i([m^j]_{\sim+}) = [n^j]_{\sim+}$. Consequently, $\pi' \cdot \langle i \rangle \in \mathcal{P}_{\bar{g}}([n^j]_{\sim+})$ holds.

Similarly, we also show (3) by induction on the length of π . If $\pi = \langle \rangle$, then we have $\langle \rangle \in \mathcal{P}_{\bar{g}}^a(N)$, i.e. $N = \bar{\tau}$. Since, by assumption, $\overline{\text{lab}}(\bar{\tau}) \in \Sigma$ holds, there is some $j \in \{1, 2\}$ such that $\text{lab}^j(r^j) \in \Sigma$. Moreover, we clearly have $\langle \rangle \in \mathcal{P}_{g_j}^a(r^j)$. If $\pi = \pi' \cdot \langle i \rangle$, then we have $\pi' \cdot \langle i \rangle \in \mathcal{P}_{\bar{g}}^a(N)$. Let $N' = \text{node}_{\bar{g}}(\pi')$. Since $\pi' \cdot \langle i \rangle$ is acyclic in \bar{g} , so is π' , i.e. $\pi' \in \mathcal{P}_{\bar{g}}^a(N')$. Moreover, we have that $\overline{\text{suc}}_i(N')$ is defined, i.e. $\overline{\text{lab}}(N')$ is not nullary and in particular not \perp . Thus, we can apply the induction hypothesis to obtain some $n^j \in N'$ with $\text{lab}^j(n^j) \in \Sigma$ and $\pi' \in \mathcal{P}_{g_j}^a(n^j)$. Hence, according to the construction of \bar{g} , we have $\text{lab}^j(n^j) = \overline{\text{lab}}(N')$, i.e. $\text{suc}_i^j(n^j) = m^j$ is defined. Furthermore, we then get $m^j \in N$. Since $\pi' \cdot \langle i \rangle \in \mathcal{P}_{g_j}(m^j)$, it remains to be shown that $\pi' \cdot \langle i \rangle$ is acyclic in g_j . Suppose that $\pi' \cdot \langle i \rangle$ is cyclic in g_j . As π' is acyclic in g_j , this means that there is some position $\pi^* \in \mathcal{P}_{g_j}(m^j)$ with $\pi^* < \pi' \cdot \langle i \rangle$. Using (2), we obtain that $\pi^* \in \mathcal{P}_{\bar{g}}(N)$. This contradicts the assumption of $\pi' \cdot \langle i \rangle$ being acyclic in \bar{g} . Hence, $\pi' \cdot \langle i \rangle$ is acyclic.

For (4), suppose that $n^j \sim^+ m^k$ holds with $\text{lab}^j(n^j), \text{lab}^k(m^k) \in \Sigma$. From (1), we obtain $\phi_j(n^j) = \phi_k(m^k)$. Moreover, since both n^j and m^k are not labelled with \perp , we know that ϕ_j and ϕ_k are rigid in n^j and m^k , respectively, which yields the equations

$$\mathcal{P}_{g_j}^a(n^j) = \mathcal{P}_{\bar{g}}^a(\phi_j(n^j)) = \mathcal{P}_{\bar{g}}^a(\phi_k(m^k)) = \mathcal{P}_{g_k}^a(m^k).$$

Next we show that $[g_1]_{\cong}, [g_1]_{\cong} \leq_{\perp}^R [\bar{g}]_{\cong}$ holds by giving two rigid \perp -homomorphisms $\psi_j: g_j \rightarrow_{\perp} \bar{g}$, $j = 1, 2$. Define $\psi_j: N^j \rightarrow \bar{N}$ by $n^j \mapsto [n^j]_{\sim+}$. From (2) and the fact that, according to the construction of \bar{g} , $\text{lab}^j(n^j) \in \Sigma$ implies $\text{lab}^j(n^j) = \overline{\text{lab}}([n^j]_{\sim+})$, we immediately get that ψ_j is a \perp -homomorphism by applying Lemma 4.10. In order to argue that ψ_j is rigid, assume that $n^j \in N^j$ with $\text{lab}^j(n^j) \in \Sigma$. According to Lemma 5.7, it suffices to show that $\mathcal{P}_{\bar{g}}^a(\psi_j(n^j)) \subseteq \mathcal{P}_{g_j}(n^j)$. Suppose that $\pi \in \mathcal{P}_{\bar{g}}^a(\psi_j(n^j))$. Note that, by the construction of \bar{g} , $\psi_j(n^j)$ is not labelled with \perp either. Hence, we can apply (3) to obtain some $m^k \in \psi_j(n^j)$ with $\text{lab}^k(m^k) \in \Sigma$ and $\pi \in \mathcal{P}_{g_k}^a(m^k)$. By definition, $m^k \in \psi_j(n^j)$ is equivalent to $n^j \sim^+ m^k$. Therefore, we can employ (4), which yields $\mathcal{P}_{g_k}^a(m^k) = \mathcal{P}_{g_j}^a(n^j)$. Hence, $\pi \in \mathcal{P}_{g_j}^a(n^j)$.

Note that the construction of \bar{g} did not depend on \hat{g} , viz. for any other upper bound $[\hat{h}]_{\cong}$ of $[g_1]_{\cong}, [g_2]_{\cong}$, we get the same term graph \bar{g} . Hence, it is still just an arbitrary upper

bound which means that in order to show that $[\bar{g}]_{\cong}$ is the least upper bound, it suffices to show $[\bar{g}]_{\cong} \leq_{\perp}^R [\hat{g}]_{\cong}$. For this purpose, we will devise a rigid \perp -homomorphism $\psi: \bar{g} \rightarrow_{\perp} \hat{g}$. Define $\psi: \bar{N} \rightarrow \hat{N}$ by $[n^j]_{\sim+} \mapsto \phi_j(n^j)$. (1) shows that ψ is well-defined. The root condition for ψ follows from the root condition for ϕ_1 :

$$\psi(\bar{r}) = \psi([r^1]_{\sim+}) = \phi_1(r^1) = \hat{r}.$$

For the labelling condition, assume that $\overline{\text{lab}}(N) = f \in \Sigma$ for some $N \in \bar{N}$. Then there is some $n^j \in N$ with $\text{lab}^j(n^j) = f$. Therefore, the labelling condition for ϕ_j yields

$$\widehat{\text{lab}}(\psi(N)) = \widehat{\text{lab}}(\phi_j(n^j)) = \text{lab}^j(n^j) = f$$

For the successor condition, let $\overline{\text{suc}}_i(N) = N'$ for some $N, N' \in \bar{N}$. Then there is some $n^j \in N$ with $\text{suc}_i^j(n^j) \in N'$. Therefore, the successor condition for ψ follows from the successor condition for ϕ_j as follows:

$$\begin{aligned} \psi(\overline{\text{suc}}_i(N)) &= \psi(N') = \psi([\text{suc}_i^j(n^j)]_{\sim+}) = \phi_j(\text{suc}_i^j(n^j)) \\ &= \widehat{\text{suc}}_i(\phi_j(n^j)) = \widehat{\text{suc}}_i(\psi([n^j]_{\sim+})) = \widehat{\text{suc}}_i(\psi(N)) \end{aligned}$$

Finally, we show that ψ is rigid. To this end, let $N \in \bar{N}$ with $\overline{\text{lab}}(N) \in \Sigma$. That is, there is some $n^j \in N$ with $\text{lab}^j(n^j) \in \Sigma$. Recall, that we have shown that $\psi_j: g^j \rightarrow_{\perp} \bar{g}$ is rigid. That is, we have

$$\mathcal{P}_{g_j}^a(n^j) = \mathcal{P}_{\bar{g}}^a(\psi_j(n^j)) = \mathcal{P}_{\bar{g}}^a([n^j]_{\sim+}).$$

Analogously, we have $\mathcal{P}_{\bar{g}}^a(\phi_j(n^j)) = \mathcal{P}_{g_j}^a(n^j)$ as ϕ_j is rigid, too. Using this, we can obtain the following equations:

$$\mathcal{P}_{\bar{g}}^a(\psi(N)) = \mathcal{P}_{\bar{g}}^a(\psi([n^j]_{\sim+})) = \mathcal{P}_{\bar{g}}^a(\phi_j(n^j)) = \mathcal{P}_{g_j}^a(n^j) = \mathcal{P}_{\bar{g}}^a([n^j]_{\sim+}) = \mathcal{P}_{\bar{g}}^a(N)$$

Hence, ψ is a rigid \perp -homomorphism from \bar{g} to \hat{g} . \square

APPENDIX B. PROOF OF LEMMA 6.10

In this appendix, we will give the full proof of Lemma 6.10. Before we can do this we have to establish a number of technical auxiliary lemmas.

The lemma below will serve as a tool for the two lemmas that are to follow afterwards. We know that the set of retained nodes $N_{<d}^g$ contains at least all nodes at depth smaller than d due to the closure condition (T1). However, due to the closure condition (T2) also nodes at a larger depth may be included in $N_{<d}^g$. The following lemma shows that this is not possible for nodes labelled with nullary symbols:

Lemma B.1 (labelling). *Let $g \in \mathcal{G}^\infty(\Sigma)$, $\Delta \subseteq \Sigma^{(0)}$ and $d < \omega$. If $\Delta\text{-depth}(g) \geq d$, then $\text{lab}^g(n) \notin \Delta$ for all $n \in N_{<d}^g$.*

Proof. We will show that $N_{\nabla} = \{n \in N^g \mid \text{lab}^g(n) \notin \Delta\}$ satisfies the properties (T1) and (T2) of Definition 6.1 for the term graph g and depth d . Since $N_{<d}^g$ is the least such set, we then obtain $N_{<d}^g \subseteq N_{\nabla}$ and, thereby, the claimed statement.

For (T1), let $n \in N^g$ with $\text{depth}_g(n) < d$. Since $\Delta\text{-depth}(g) \geq d$, we have $\text{lab}^g(n) \notin \Delta$ and, therefore, $n \in N_{\nabla}$. For (T2), let $n \in N_{\nabla}$ and $m \in \text{Pre}_g^a(n)$. Then m cannot be labelled with a nullary symbol, a fortiori $\text{lab}^g(m) \notin \Delta$. Hence, we have $m \in N_{\nabla}$. \square

The following two lemmas are rather technical. They state that rigid Δ -homomorphisms preserve retained nodes and in a stricter sense also fringe nodes.

Lemma B.2 (preservation of retained nodes). *Let $g, h \in \mathcal{G}^\infty(\Sigma)$, $d < \omega$, $\phi: g \rightarrow_\Delta h$ rigid, and $\Delta\text{-depth}(g) \geq d$. Then $\phi(N_{<d}^g) = N_{<d}^h$.*

Proof. Let $N_\nabla = \{n \in N^g \mid \text{lab}^g(n) \notin \Delta\}$. At first we will show that $\phi(N_{<d}^g) \subseteq N_{<d}^h$. To this end, we will show that $\phi^{-1}(N_{<d}^h) \cap N_\nabla$ satisfies (T1) and (T2) of Definition 6.1 for term graph g and depth d . Since $N_{<d}^g$ is the least such set, we then obtain $N_{<d}^g \subseteq \phi^{-1}(N_{<d}^h) \cap N_\nabla$ and, a fortiori, $N_{<d}^g \subseteq \phi^{-1}(N_{<d}^h)$ which is equivalent to $\phi(N_{<d}^g) \subseteq N_{<d}^h$.

For (T1), let $n \in N^g$ with $\text{depth}_g(n) < d$. Because $\Delta\text{-depth}(g) \geq d$, we know that $\text{lab}^g(n) \notin \Delta$, which means by Lemma 6.7 that we also have $\text{depth}_h(\phi(n)) < d$. Hence, $\phi(n) \in N_{<d}^h$ by (T1). Since $\text{lab}^g(n) \notin \Delta$, we thus have $n \in \phi^{-1}(N_{<d}^h) \cap N_\nabla$.

For (T2), let $n \in \phi^{-1}(N_{<d}^h) \cap N_\nabla$. That is, we have $\phi(n) \in N_{<d}^h$ and $\text{lab}^g(n) \notin \Delta$. Hence, by (T2), it holds that $\text{Pre}_h^a(\phi(n)) \subseteq N_{<d}^h$. We have to show now that $\text{Pre}_g^a(n) \subseteq \phi^{-1}(N_{<d}^h) \cap N_\nabla$. Let $m \in \text{Pre}_g^a(n)$. That is, there is some $\pi \cdot \langle i \rangle \in \mathcal{P}_g(n)$ with $\pi \in \mathcal{P}_g(m)$. As $\text{lab}^g(n) \notin \Delta$ and ϕ is rigid, we know that ϕ is rigid in n . Consequently, $\pi \cdot \langle i \rangle \in \mathcal{P}_h^a(\phi(n))$. Moreover, we have $\pi \in \mathcal{P}_h(\phi(m))$ by Lemma 4.10. Hence, $\phi(m) \in \text{Pre}_h^a(\phi(n))$ and, therefore, $\phi(m) \in N_{<d}^h$ by (T2). Additionally, as m has a successor in g , it cannot be labelled with a symbol in Δ . Hence, $m \in \phi^{-1}(N_{<d}^h) \cap N_\nabla$.

In order to prove the converse inclusion $\phi(N_{<d}^g) \supseteq N_{<d}^h$, we will show that $\phi(N_{<d}^g)$ satisfies (T1) and (T2) for term graph h and depth d . This will prove the abovementioned inclusion since $N_{<d}^h$ is the least such set.

For (T1), let $n \in N^h$ with $\text{depth}_h(n) < d$. By Lemma 6.5, there is some $m \in N^g$ with $\text{depth}_g(m) < d$ and $\phi(m) = n$. Hence, according to (T1), we have $m \in N_{<d}^g$ and, therefore, $n \in \phi(N_{<d}^g)$.

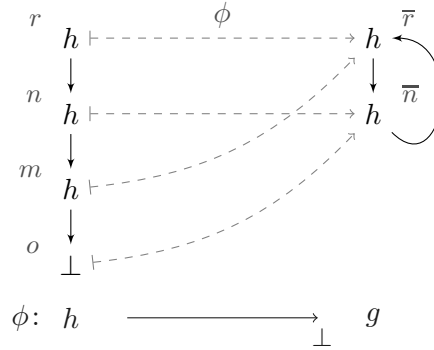
For (T2), let $n \in \phi(N_{<d}^g)$. That is, there is some $m \in N_{<d}^g$ with $\phi(m) = n$. By (T2), we have $\text{Pre}_g^a(m) \subseteq N_{<d}^g$. We must show that $\text{Pre}_h^a(n) \subseteq \phi(N_{<d}^g)$. Let $n' \in \text{Pre}_h^a(n)$. That is, there is some $\pi \cdot \langle i \rangle \in \mathcal{P}_h(n)$ with $\pi \in \mathcal{P}_h(n')$. Since $m \in N_{<d}^g$, we have $\text{lab}^g(m) \notin \Delta$ by Lemma B.1. Consequently, ϕ is rigid in m which yields that $\pi \cdot \langle i \rangle \in \mathcal{P}_g(m)$. Note that then also $\pi \in \mathcal{P}(g)$. Let $m' = \text{node}_g(\pi)$. Thus, $m' \in \text{Pre}_g^a(m)$ and, therefore, $m' \in N_{<d}^g$ according to (T2). Moreover, because $\pi \in \mathcal{P}_g(m') \cap \mathcal{P}_h(n')$, we are able to obtain from Lemma 4.10 that $\phi(m') = n'$. Hence, $n' \in \phi(N_{<d}^g)$. \square

Lemma B.3 (preservation of fringe nodes). *Let $g, h \in \mathcal{G}^\infty(\Sigma)$, $\phi: g \rightarrow_\Delta h$ rigid, $0 < d < \omega$, $\Delta\text{-depth}(g) \geq d$, $n \in N^g$, and $0 \leq i < \text{ar}_g(n)$. Then $n^i \in N_{=d}^g$ iff $\phi(n)^i \in N_{=d}^h$.*

Proof. Note that, by Lemma B.1, we have that $\text{lab}^g(n) \notin \Delta$ for all nodes $n \in N_{<d}^g$. Additionally, by Lemma B.2, we obtain $\phi(N_{<d}^g) = N_{<d}^h$ and, therefore, according to the labelling condition for ϕ , we get that $\text{lab}^h(n) \notin \Delta$ for all $n \in N_{<d}^h$.

At first we will show the “only if” direction. To this end, let $n^i \in N_{=d}^g$. By definition, we then have $\text{depth}_g(n) \geq d-1$. Hence, by Lemma 6.7, $\text{depth}_h(\phi(n)) \geq d-1$. Furthermore, we have that $\text{suc}_i^g(n) \notin N_{<d}^g$ or $n \notin \text{Pre}_g^a(\text{suc}_i^g(n))$. We show now that in either case we can conclude $\phi(n)^i \in N_{=d}^h$.

Let $\text{suc}_i^g(n) \notin N_{<d}^g$. If we have $\text{suc}_i^h(\phi(n)) \notin N_{<d}^h$, then $\phi(n)^i \in N_{=d}^h$. So suppose $\text{suc}_i^h(\phi(n)) \in N_{<d}^h$. Since $N_{<d}^h = \phi(N_{<d}^g)$, according to Lemma B.2, we find some $m \in N_{<d}^g$ with $\phi(m) = \text{suc}_i^h(\phi(n))$. However, since $\text{suc}_i^g(n) \notin N_{<d}^g$, we know that $m \neq \text{suc}_i^g(n)$.


 FIGURE 11. Fringe nodes and rigid \perp -homomorphisms.

We now show $\phi(n) \notin \text{Pre}_h^a(\text{suc}_i^h(\phi(n)))$ by showing that $\pi \cdot \langle i \rangle \notin \mathcal{P}_h^a(\text{suc}_i^h(\phi(n)))$ whenever $\pi \in \mathcal{P}_h^a(\phi(n))$:

$$\begin{aligned}
 \pi \in \mathcal{P}_h^a(\phi(n)) &\iff \pi \in \mathcal{P}_g^a(n) && (\phi \text{ is rigid in } n) \\
 &\implies \pi \cdot \langle i \rangle \notin \mathcal{P}_g^a(m) && (m \neq \text{suc}_i^g(n)) \\
 &\iff \pi \cdot \langle i \rangle \notin \mathcal{P}_h^a(\phi(m)) && (\phi \text{ is rigid in } m) \\
 &\iff \pi \cdot \langle i \rangle \notin \mathcal{P}_h^a(\text{suc}_i^h(\phi(n))) && (\phi(m) = \text{suc}_i^h(\phi(n)))
 \end{aligned}$$

Together with $\text{depth}_h(\phi(n)) \geq d - 1$, this implies that $\phi(n)^i \in N_{=d}^h$.

Let $n \notin \text{Pre}_g^a(\text{suc}_i^g(n))$. If $\phi(n) \notin \text{Pre}_h^a(\text{suc}_i^h(\phi(n)))$, then $\phi(n)^i \in N_{=d}^h$. So suppose that $\phi(n) \in \text{Pre}_h^a(\text{suc}_i^h(\phi(n)))$. Hence, $\phi(n) \in \text{Pre}_h^a(\phi(\text{suc}_i^g(n)))$ as ϕ is homomorphic in n . If $\text{lab}^g(\text{suc}_i^g(n)) \notin \Delta$, then ϕ is rigid in $\text{suc}_i^g(n)$ and we would also get $n \in \text{Pre}_g^a(\text{suc}_i^g(n))$ which contradicts the assumption. Hence, $\text{lab}^g(\text{suc}_i^g(n)) \in \Delta$ and, therefore, $\text{suc}_i^g(n) \notin N_{<d}^g$ according to Lemma B.1. Thus, we can employ the argument for this case that we have already given above.

We now turn to the converse direction. For this purpose, let $\phi(n)^i \in N_{=d}^h$. Then $\text{depth}_h(\phi(n)) \geq d - 1$ and, consequently $\text{depth}_g(n) \geq d - 1$ by Lemma 6.7. Additionally, we also have $\text{suc}_i^h(\phi(n)) \notin N_{<d}^h$ or $\phi(n) \notin \text{Pre}_h^a(\text{suc}_i^h(\phi(n)))$. Again we will show that in either case we can conclude $n^i \in N_{=d}^g$.

If $\text{suc}_i^h(\phi(n)) \notin N_{<d}^h$, then $\phi(\text{suc}_i^g(n)) \notin N_{<d}^h$ and, therefore, $\phi(\text{suc}_i^g(n)) \notin N_{<d}^g$ according to Lemma B.2. Consequently, $\text{suc}_i^g(n) \notin N_{<d}^g$ which implies that $n^i \in N_{=d}^g$.

Let $\phi(n) \notin \text{Pre}_h^a(\text{suc}_i^h(\phi(n)))$. If $n \notin \text{Pre}_g^a(\text{suc}_i^g(n))$, then we get $n^i \in N_{=d}^g$ immediately. So assume that $n \in \text{Pre}_g^a(\text{suc}_i^g(n))$. If $\text{lab}^g(\text{suc}_i^g(n)) \notin \Delta$, then ϕ would be rigid in $\text{suc}_i^g(n)$. Thereby, we would get $\phi(n) \in \text{Pre}_h^a(\phi(\text{suc}_i^g(n)))$ which contradicts the assumption. Hence, $\text{lab}^g(\text{suc}_i^g(n)) \in \Delta$. According to Lemma B.1, we then have $\text{suc}_i^g(n) \notin N_{<d}^g$ and, therefore, $n^i \in N_{=d}^g$. \square

The above lemma depends on the peculiar definition of fringe nodes – in particular those fringe nodes that are due to the condition

$$\text{depth}_g(n) \geq d - 1 \text{ and } n \notin \text{Pre}_g^a(\text{suc}_i^g(n)).$$

Recall that this condition produces a fringe node for each edge from a retained node that closes a cycle. Let us have a look at the term graph g depicted in Figure 11. The rigid

truncation $g \ddagger 2$ of g is shown in Figure 7a. If the abovementioned alternative condition for fringe nodes would not be present, then the set $N_{=2}^g$ would be empty (and, thus, $g \ddagger 2 = g$). Then, however, the rigid \perp -homomorphism ϕ illustrated in Figure 11 would violate Lemma B.3. Since the node m is cut off from h in the truncation $h \ddagger 2$, there is a fringe node n^0 in $h \ddagger 2$. On the other hand, there would be no fringe node \bar{n}^0 in $g \ddagger 2$ if not for the alternative condition above.

Lemma 6.10 (\leq_{\perp}^R and rigid truncation). *Given $g, h \in \mathcal{G}^{\infty}(\Sigma_{\perp})$ and $d < \omega$ with $g \leq_{\perp}^R h$ and $\perp\text{-depth}(g) \geq d$, we have that $g \ddagger d \cong h \ddagger d$.*

Proof of Lemma 6.10. For $d = 0$, this is trivial. So assume $d > 0$. Since $g \leq_{\perp}^R h$, there is a rigid \perp -homomorphism $\phi: g \rightarrow_{\perp} h$. Define the function ψ as follows:

$$\begin{aligned} \psi: N^{g \ddagger d} &\rightarrow N^{h \ddagger d} \\ N_{<d}^g \ni n &\mapsto \phi(n) \\ N_{=d}^g \ni n^i &\mapsto \phi(n)^i \end{aligned}$$

At first we have to argue that ψ is well-defined. For this purpose, we first need that $\phi(N_{<d}^g) \subseteq N^{g \ddagger d}$. Lemma B.2 confirms this. Secondly, we need that $n^i \in N_{=d}^g$ implies $\phi(n)^i \in N^{g \ddagger d}$. This is guaranteed by Lemma B.3.

Next we show that ψ is a homomorphism from $g \ddagger d$ to $h \ddagger d$. The root condition is inherited from ϕ as $r^{g \ddagger d} \in N_{<d}^g$. Note that, according to Lemma B.1, we have $\text{lab}^g(n) \in \Sigma$ for all $n \in N_{<d}^g$. Hence, ϕ is homomorphic in $N_{<d}^g$ which means that the labelling condition for nodes in $N_{<d}^g$ is also inherited from ϕ . For nodes $n^i \in N_{=d}^g$, we have $\text{lab}^{g \ddagger d}(n^i) = \perp$. Since, by definition, $\psi(n^i) \in N_{=d}^h$, we can conclude $\text{lab}^{h \ddagger d}(\psi(n^i)) = \perp$.

The successor condition is trivially satisfied by nodes in $N_{=d}^g$ as they do not have any successors. Let $n \in N_{<d}^g$ and $0 \leq i < \text{ar}_{g \ddagger d}(n)$. We distinguish two cases: At first assume that $n^i \notin N_{=d}^g$. Hence, $\text{suc}_i^{g \ddagger d}(n) = \text{suc}_i^g(n) \in N_{<d}^g$. Since, by Lemma B.3, also $\phi(n)^i \notin N_{=d}^h$, we additionally have $\text{suc}_i^{h \ddagger d}(\phi(n)) = \text{suc}_i^h(\phi(n))$. Hence, using the successor condition for ϕ , we can reason as follows:

$$\psi(\text{suc}_i^{g \ddagger d}(n)) = \psi(\text{suc}_i^g(n)) = \phi(\text{suc}_i^g(n)) = \text{suc}_i^h(\phi(n)) = \text{suc}_i^{h \ddagger d}(\phi(n)) = \text{suc}_i^{h \ddagger d}(\psi(n))$$

If, on the other hand, $n^i \in N_{=d}^g$, then $\text{suc}_i^{g \ddagger d}(n) = n^i$. Moreover, since then $\phi(n)^i \in N_{=d}^h$ by Lemma B.3, we have $\text{suc}_i^{h \ddagger d}(\phi(n)) = \phi(n)^i$, too. Hence, we can reason as follows:

$$\psi(\text{suc}_i^{g \ddagger d}(n)) = \psi(n^i) = \phi(n)^i = \text{suc}_i^{h \ddagger d}(\phi(n)) = \text{suc}_i^{h \ddagger d}(\psi(n))$$

This shows that ψ is a homomorphism. Note that, according to Lemma 5.6, ϕ is injective in $N_{<d}^g$. Then also ψ is injective in $N_{<d}^g$. For the same reason, ψ is also injective in $N_{=d}^g$. Moreover, we have $\psi(N_{<d}^g) \subseteq N_{<d}^h$ and $\psi(N_{=d}^g) \subseteq N_{=d}^h$, i.e. $\psi(N_{<d}^g) \cap \psi(N_{=d}^g) = \emptyset$. Hence, ψ is injective which implies, by Lemma 4.12, that ψ is an isomorphism from $g \ddagger d$ to $h \ddagger d$. \square